

SATAKUNNAN AMMATTIKORKEAKOULU



Jussi Männistö

2008

KUORMITUSTIETOKANNAN JA LOADSDB-OHJELMISTON
KEHITTÄMINEN

Tekniikka Rauma

Tietotekniikan koulutusohjelma

KUORMITUSTIETOKANNAN JA LOADSDB-OHJELMISTON KEHITTÄMINEN

Männistö, Jussi
Satakunnan ammattikorkeakoulu
Tekniikka Rauma
Tietotekniikan koulutusohjelma
Työn tilaaja: Teollisuuden Voima Oyj
Työn valvoja yrityksessä: Paul Smeekes
Huhtikuu 2008
Ohjaaja: Mikko Javanainen
UDK: 004.65; 004.43
Sivumäärä: 41

Asiasanat: ohjelmistotekniikka, tietokannat, Visual Basic, Access, SQL

Tämä opinnäytetyö tehtiin Teollisuuden Voima Oyj:lle. Sen aiheena oli kuormitus-tietokannan edelleen kehittäminen sekä uusien ominaisuuksien lisääminen LoadsDb-ohjelmistoon. Sekä kuormitustietokanta että LoadsDb-ohjelmisto ovat osa PAMS-nimistä järjestelmää. PAMS on lyhenne englanninkielisistä sanoista The TVO Piping and Component Analysis and Monitoring System.

Koska olin jo aiemmin tutustunut kyseiseen järjestelmään ja kehittänyt LoadsDb-ohjelmistoa kesätyönä, saatoinkin aloittaa uusien ominaisuuksien kehittämisen välittömästi. Tietokannan ylläpitoon käytettiin Microsoft Access -tietokantaohjelmistoa ja sovelluskehittimenä toimi Microsoft Visual Basic 6.

Uudet ominaisuudet annettiin tuoreeltaan käyttäjien testattaviksi. Näin käyttäjien arvokas palaute saatiin lähes välittömästi, ja tätä kautta mahdolliset virheet voitiin poistaa ohjelmistosta mahdollisimman nopeasti.

Opinnäytetyön tuloksena kuormitustietokanta kehittyi askeleen eteenpäin ja LoadsDb-ohjelmisto on nyt entistä monipuolisempi. Vaikka opinnäytetyö onkin valmis, kuormitustietokannan ja LoadsDb-ohjelmiston kehittäminen jatkuu edelleen.

DEVELOPMENT OF LOADING DATABASE AND LOADSDB SOFTWARE

Männistö, Jussi
Satakunta University of Applied Sciences
School of Technology Rauma
Information Technology
Commissioned by Teollisuuden Voima Oyj
Supervisor: Paul Smeeke
April 2008
Tutor: Mikko Javanainen
UDC: 004.65; 004.43
Number of pages: 41

Keywords: software technology, databases, Visual Basic, Access, SQL

This Bachelor's thesis was commissioned by Teollisuuden Voima Oyj. The purpose of the project was to improve the loading database and to add new features to the software called LoadsDb. They are both part of a system called PAMS, which stands for the TVO Piping and Component Analysis and Monitoring System.

Having developed the LoadsDb and other software included in the PAMS earlier, the author was familiar with the subject and was able to start developing the software right away. The LoadsDb software was developed with Microsoft Visual Basic 6 and the loading database was maintained with Microsoft Access.

The newly developed features were immediately tested by the end-users. Through this method, valuable user feedback was obtained constantly, and possible defects were amended instantly.

As a result, the loading database was improved and new features were added to the LoadsDb. Although this thesis has been completed, further development of the loading database and the LoadsDb software still continues.

SYMBOLI- JA TERMILUETTELO

ADO	Lyhenne sanoista ActiveX Data Objects. ADO on Microsoftin kehittämä kokoelma Component Object Model (COM) objekteja tiedon hallintaan. ADO toimii yhteytenä ohjelmointikielien ja OLE DB:n välillä.
Ajonaikainen kirjasto	Jaettu kirjasto tai ajonaikainen kirjasto termillä viitataan yleisesti käyttöjärjestelmien tapaan jakaa ohjelmakoodia tai dataa useiden ohjelmien kesken. Ajonaikaisten kirjastojen etuna on keskusmuistin ja levytilan säästyminen.
API	Application Programming Interface, suomeksi ohjelmointirajapinta, on käyttöliittymä jolla eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään.
Dll-kirjasto	Puhuttaessa dll-kirjastoista viitataan erityisesti Microsoft Windows-ympäristössä esiintyvään tapaan toteuttaa ajonaikainen kirjasto. Dll on lyhenne englannin kielen sanoista dynamic link library.
Microsoft Access	Microsoft Office -tuoteperheeseen kuuluva tietokantojen hallintaohjelma. Accessilla luodut tietokannat ovat relaatiotietokantoja.
Microsoft Visual Basic 6	Microsoftin kehittämän BASIC-sukuisen ohjelmointikielen versio kuusi. On myös edellä mainittuun ohjelmointikielen perustuvien ohjelmien kehitysympäristön nimi.
OLE DB	Microsoftin kehittämä API erityyppisen, mutta yhdenmukaisesti varastoidun tiedon hallintaan.
Relaatiotietokanta	Tauluihin perustuva tietokantamalli, joka koostuu rivien sekä sarakkeiden muodostamista tauluista ja taulujen välisistä yhteyksistä.
SQL	Lyhenne sanoista structured query language. IBM:n kehittämä standardoitu kyselykieli, jolla on mahdollista tehdä erilaisia operaatioita, kuten tiedon haku, lisäys ja poisto, relaatiotietokantaan.
Transientti	Suureen, kuten jännite, virta tai kuormitus, muutos ajassa.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

SYMBOLI- JA TERMI LUETTELO

1	JOHDANTO.....	7
2	PAMS.....	8
2.1	Yleistä PAMS:sta.....	8
2.2	PAMS:n rakenne.....	9
2.2.1	Järjestelmän osat.....	9
2.2.2	Ulkoalue.....	10
2.2.3	Neutral File Area.....	10
2.2.4	Tietokanta-alue.....	10
3	KUORMITUSTIETOKANTA JA LOADSDB-OHJELMISTO.....	14
3.1	Kuormitustietokanta.....	14
3.1.1	Kuormitustietokannan sisältö.....	14
3.1.2	Kuormitustietokannan rakenne.....	14
3.2	LoadsDb-ohjelmisto.....	15
3.2.1	LoadsDb-ohjelmiston tehtävä.....	15
3.2.2	LoadsDb-ohjelmiston päänäky.....	15
4	TYÖKALUT.....	16
4.1	Sovelluskehitin.....	16
4.2	Tietokantasovellus.....	17
4.3	Piirto-ohjelma.....	17
4.4	Muut ohjelmat.....	18
5	SPEKTRIT JA KERROSTUMAT.....	18
5.1	Ominaisuuden tarkoitus.....	18
5.2	Muutokset tietokantaan.....	18
5.3	Ominaisuuden ohjelmallinen toteutus.....	19
5.3.1	Uusi kenttä.....	19
5.3.2	Näytettävien käyrien lukumäärän valinta.....	20
5.3.3	Spektrien ja kerrostumien visualisointi.....	20
6	HAKULOMAKKEEN KEHITTÄMINEN.....	23
6.1	Hakulomake.....	23
6.2	Hakukenttien suodatus.....	23

6.3	Uusi hakukenttä: Plant	24
6.4	Muut muutokset	24
7	HAKUTULOSTEN ASKELLUS	26
7.1	Ominaisuuden tarkoitus	26
7.2	Ominaisuuden toteutus	26
7.2.1	Yleistä ominaisuuden toteutuksesta	26
7.2.2	Hakutulosten tallennus	26
7.2.3	Askelluspainikkeet	27
7.2.4	Askelluksen lukitus	28
7.2.5	Askelluksen vapautus	29
7.2.6	Toimenpiteet askelluksen lukituksen ja vapautuksen yhteydessä.....	31
8	ALUEKUVAN NÄYTTÄMINEN	32
8.1	Ominaisuuden tarkoitus	32
8.2	Ominaisuuden toteutus	32
8.2.1	Yleistä ominaisuuden toteutuksesta	32
8.2.2	Saatavilla olevien dokumenttien lukumäärän tarkastus	33
8.2.3	Komentopainikkeen toteutus.....	35
8.2.4	Dokumentin avaaminen.....	36
9	YHTEENVETO	38
	LÄHTEET.....	39
	LIITTEET	

1 JOHDANTO

Opinnäytetyö tehtiin Teollisuuden Voima Oyj:lle ja työn valvojana yrityksen puolelta toimi lujuuslaskentapäällikkö Paul Smeekes. Hänen lisäksään työtä ohjasi ohjelmistokonsultti Olli Kuuluvainen ja tärkeää käyttäjäpalautetta toimitti putkistokoordinaattori Niila Kata.

Työn aiheena oli PAMS-nimiseen kokonaisuuteen kuuluvan kuormitustietokannan sekä kuormitustietokannan tietojen syöttämiseen ja muokkaamiseen tarkoitettun LoadsDb-ohjelmiston kehittäminen. Ennen työn aloittamista käydyssä kokouksessa työn aihetta tarkennettiin ja sovittiin sen alustavasti käsittävän kuuden sillä hetkellä tärkeimmiksi arvioidun ominaisuuden kehittämisen. Lisäksi sovittiin, että opinnäytetyön aihetta voitaisiin rajata uudelleen, mikäli tähän olisi tarvetta. Tällaiseksi tarpeeksi katsottiin esimerkiksi se, että jonkin alkuperäiseen suunnitelmaan kuulumattoman ominaisuuden kehittäminen havaittaisiin näitä suunniteltuja ominaisuuksia tärkeämmäksi. Toisena mahdollisena syynä opinnäytetyön uudelleen rajaukseen pidettiin myös aiheen laajuutta sekä tiukkaa aikataulua. /1/

Ensimmäinen näistä kuudesta ominaisuudesta oli mahdollisuus asettaa syötettyjä transientteja peräkkäin, eli muodostaa näistä peräkkäisistä transienteista syklejä. Toinen ominaisuus oli mahdollistaa spektrikuormien ja kerrostumien syöttäminen sekä näiden visualisoinnin kehittäminen. Syötettyjen tietojen tarkastelua helpottavan hakulomakkeen kehittäminen oli kolmas ominaisuus.

Edelliseen läheisesti liittyvä neljäs ominaisuus oli hakulomakkeen hakutulosten pohjalta tapahtuvan askelluksen toteutus. Viides ominaisuus oli mahdollisuus tarkastella kunkin transientin lähdemateriaaliin kuuluvaa aluekuvaa. Kuudes ja viimeinen ominaisuus oli päivittää dll-pohjainen liityntä, jonka kautta muiden ohjelmien on mahdollista saada kuormitustietokannan sisältämää tietoa.

Arvio siitä, että opinnäytetyön aihetta olisi mahdollisesti myöhemmin rajattava uudelleen, osui oikeaan. Kunkin ominaisuuden sisällön tarkentuessa havaittiin, että joikaista kuudesta suunnitellusta ominaisuudesta ei olisi mahdollista toteuttaa käytettävissä olevassa ajassa. Niinpä edellä mainituista ominaisuuksista ensimmäinen ja kuudes, eli syklit sekä dll-pohjainen liityntä, päätettiin rajata opinnäytetyön ulkopuolelle ja siirtää niiden kehitys myöhempään ajankohtaan.

Kuormitustietokanta itsessään on melko monimutkainen ja LoadsDb-ohjelmisto sisältää kattavan määrän ominaisuuksia. On kuitenkin muistettava, että nämä kaksi ovat vain osia suuremmasta kokonaisuudesta. Onkin kohtuullista todeta, että ymmärtääkseen kuormitustietokannan tehtävää ja toimintaa, on syytä tutustua PAMS:iin kokonaisuudessaan.

2 PAMS

2.1 Yleistä PAMS:sta

PAMS on lyhenne englanninkielen sanoista The TVO Piping and Component Analysis and Monitoring System, joka on vapaasti käännettynä suomeksi TVO:n putkistojen sekä komponenttien analysointi- ja tarkastelujärjestelmä. Tämän järjestelmän tarkoituksena on toimia apuna Teollisuuden Voima Oyj:n laitossyksiköitä koskevan tiedon syöttämisessä ja tarkastelussa.

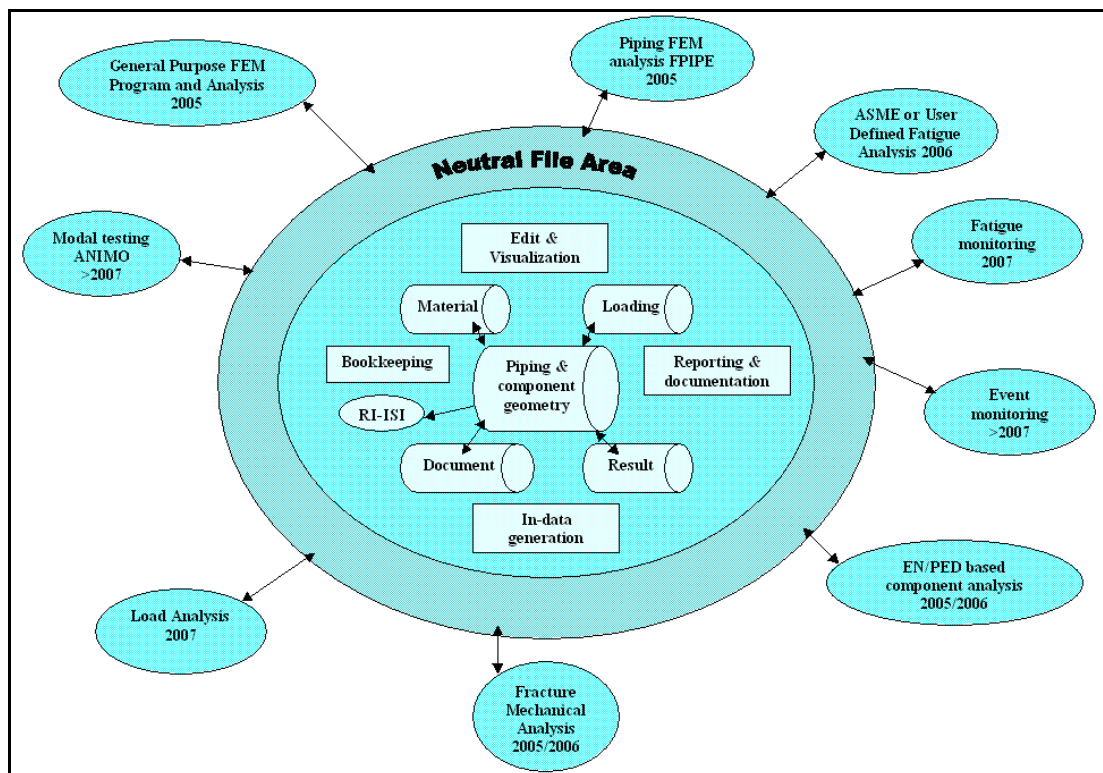
Tätä tietoa on erittäin runsaasti ja se on sisällöltään monipuolista. Järjestelmän sisältämää tietoa ovat esimerkiksi erilaiset laitoksien tiettyihin osiin kohdistuvat transienitit ja vaikkapa laitoksella tietyssä kohtaa sijaitsevan putken materiaali. Lisäksi yhtenä järjestelmän kantavana ajatuksena on, että jokaisella järjestelmään syötettävällä tiedolla on oltava luotettava lähde, joka dokumentoidaan järjestelmään. Näin tiedon alkuperä voidaan jäljittää vielä vuosia myöhemmin.

Ensisijaisesti tiedot ovat katseltavissa ja muokattavissa kunkin tietotyypin tarkoitukseen rakennetussa ohjelmassa. Tarkoituksena on kuitenkin, että kaikki tämä tieto olisi tarvittaessa saatavilla myös muissa järjestelmään kuuluvissa ohjelmistoissa. Tämän vuoksi järjestelmän eri osat sisältävät muille osille tarkoitettua tietoa kyselyä varten luodut liittynät.

2.2 PAMS:n rakenne

2.2.1 Järjestelmän osat

PAMS sisältää useita eri osia, jotka voidaan kuitenkin jakaa kolmeen pääryhmään. Nämä ryhmät ovat tietokanta-alue, Neutral File Area -niminen rajapinta-alue sekä ulkoalue. Eri osat ja niiden väliset yhteydet on esitetty kuvassa 1.



Kuva 1. PAMS:n osat ja osien väliset yhteydet.

2.2.2 Ulkoalue

Ulkoalue sisältää kahta päätyyppiä olevia ohjelmia. Ensimmäistä päätyyppiä ovat kaupalliset, järjestelmän sisältämien tietojen analysointiin tarkoitetut ohjelmat, kuten esimerkiksi FPIPE. Toinen päätyyppi käsittää useita käyttötarkoitukseltaan erilaisia, juuri järjestelmää varten räätälöityjä ohjelmia. Näillä ohjelmilla on mahdollista suorittaa erilaisia, ajoiksi kutsuttuja, tietokanta-alueen tietoa hyödyntäviä tiedon analysointitoimenpiteitä.

2.2.3 Neutral File Area

Neutral File Area -niminen rajapinta-alue on tarkoitettu tietokanta-alueen ja ulkoalueen väliseksi yhteydeksi. Sen tarkoituksena on toimia apuna tiedon muuntamisessa kullekin ohjelmistolle sopivaan muotoon. Tällä tarkoitetaan sitä, että se luo tietokanta-alueen tiedoista kullekin ulkoalueen ohjelmalle sopivan syöttötiedoston, jonka perusteella ulkoalueen ohjelma suorittaa valitut toimenpiteet. Tämän jälkeen on rajapinta-aluetta käyttäen mahdollista viedä ulkoalueen ohjelmien tuottamaa tietoa takaisin tietokanta-alueelle.

2.2.4 Tietokanta-alue

2.2.4.1 Tietokanta-alueen osat

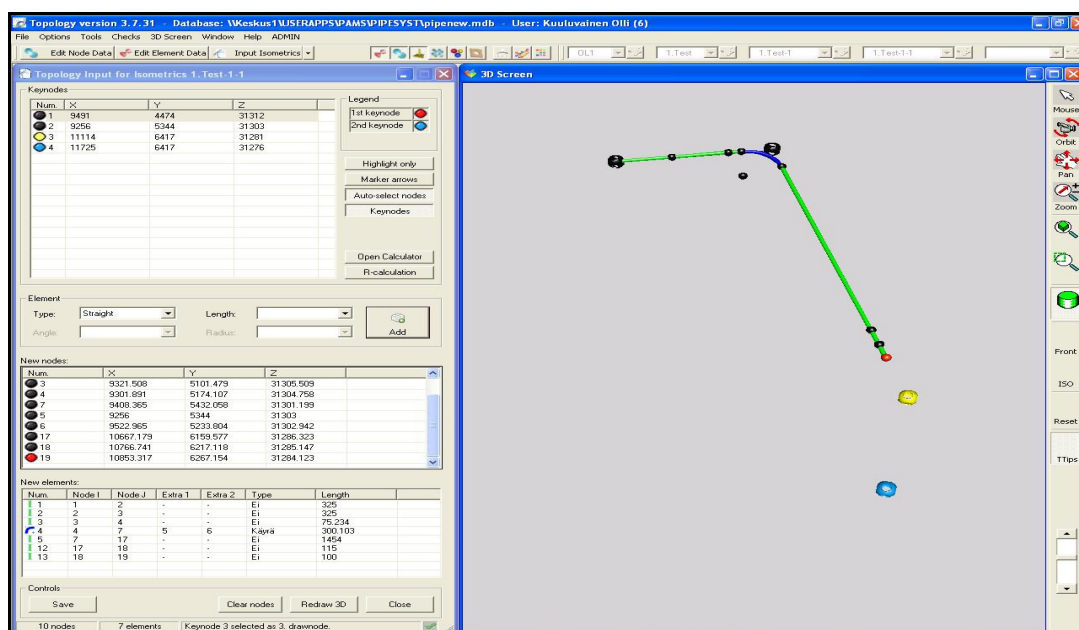
Tietokanta-alue sisältää useita toisiinsa liitettyjä Microsoft Access -pohjaisia tietokantoja sekä näiden tietokantojen sisältämän tiedon hallintaan tarkoitettuja ohjelmistoja. Alueen tärkeimmät osat ovat putkisto- ja komponenttitietokanta, materiaalitietokanta, tulostietokanta, dokumenttitietokanta sekä kuormitustietokanta.

2.2.4.2 Putkisto- ja komponenttitietokanta

Putkisto- ja komponenttitietokanta yhdessä Topology-ohjelmiston kanssa muodostavat PAMS:n keskeisimmän osan. Tietokannan sisältämää tietoa ovat muun muassa

laitosyksiköiden topologia, komponenteissa käytetyt materiaalit sekä alueisiin kohdistuvat kuormitukset. Topology sisältää useita työkaluja tämän tiedon visualisointiin.

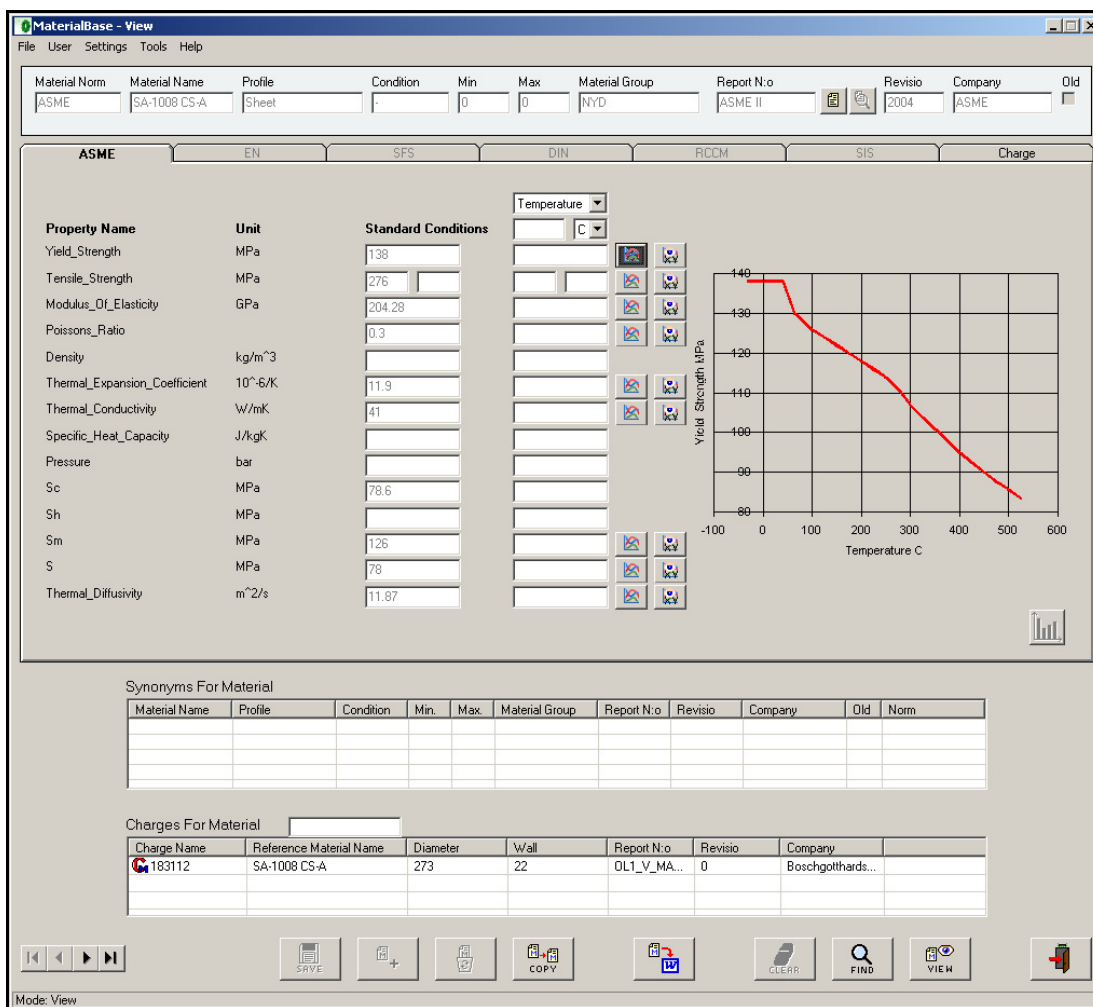
Sen avulla on esimerkiksi mahdollista tuottaa putkistojen geometriasta kolmiulotteinen malli, jossa putkistossa kulkevia virtauksia on kuvattu liukuvärein. Kuvassa 2 on esitetty Topology-ohjelmiston näkymä, jossa vasemmalla sijaitsevat tietojen syöttämiseen tarkoitetut kentät ja oikealla on näkyvissä jo syötetyistä tiedoista muodostettu putkiston kolmiulotteinen visualisointi.



Kuva 2. Topology-ohjelmiston näkymä, jossa kolmiulotteinen visualisointi.

2.2.4.3 Materiaalitietokanta

Materiaalitietokanta sisältää yksityiskohtaisia tietoja laitosyksiköillä käytetyistä materiaaleista. Tällaisia tietoja ovat esimerkiksi materiaalin sulamispiste, lämpölaajenemiskerroin, tiheys ja kimmokerroin. Tietokannan hallintaan käytetään MaterialBase-ohjelmistoa. MaterialBase-ohjelmiston päänäkymä on esitetty kuvassa 3.



Kuva 3. MaterialBase-ohjelmiston päänäkymä.

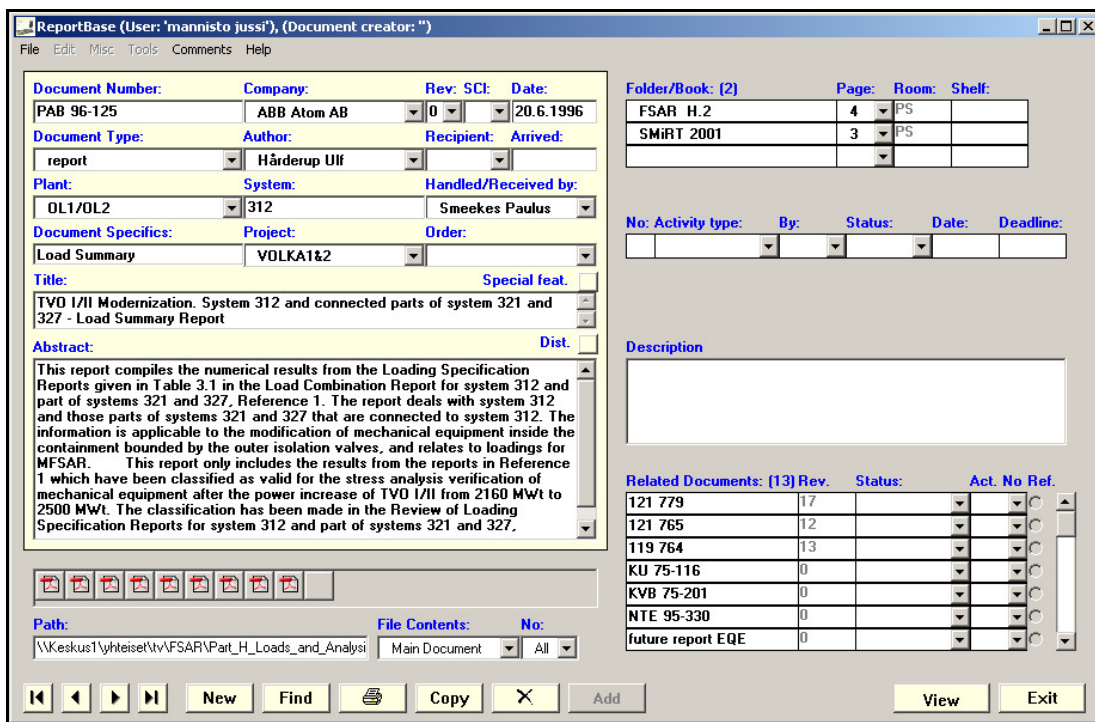
2.2.4.4 Tulostietokanta

Ulkoalueen ohjelmien tuottama tieto löytyy tulostietokannasta. Muista tietokannoista poiketen tulostietokanta ei kuitenkaan ole staattinen ja pysyvä, vaan ajon yhteydessä tulostietokanta muodostetaan dynaamisesti joka kerta uudelleen. Ajon jälkeen tulostietokannasta on mahdollista siirtää ajon tuloksena syntynyttä tietoa takaisin tietokanta-alueen tietokantoihin Neutral File Area rajapinta-alueen läpi.

2.2.4.5 Dokumenttitietokanta

Dokumenttitietokannassa on tieto kaikkien muiden tietokantojen sisältämän tiedon lähteestä. Näin on mahdollista varmistaa tietojen alkuperä ja luotettavuus jopa vuosia myöhemmin. Yleisimmin tieto on sähköisessä muodossa, jolloin dokumenttitietokan-

ta sisältää linkin esimerkiksi pdf- tai doc-formaatissa olevaan tiedostoon. Dokumenttitietokannan hallintaan tarkoitettu ohjelmisto on nimeltään ReportBase, jonka päänäköymä on esitetty kuvassa 4.



Kuva 4. ReportBase-ohjelmiston päänäköymä.

2.2.4.6 Kuormitustietokanta

Viimeinen, ja tämän opinnäytetyön kannalta tärkein kokonaisuus on kuormitustietokanta, sekä sen hallintaan tarkoitettu LoadsDb-ohjelmisto. Tämä kokonaisuus on aiheena luvussa 3. /2/

3 KUORMITUSTIETOKANTA JA LOADSDB-OHJELMISTO

3.1 Kuormitustietokanta

3.1.1 Kuormitustietokannan sisältö

Kuormitustietokanta sisältää runsaasti käyttötarkoitukseltaan ja sisällöltään vaihtelevaa tietoa. Sen pääsisältöä ovat kuitenkin erilaiset laitosyksiköillä esiintyvät transientit. Näihin transientteihin liitettyä tietoa ovat esimerkiksi transientin perustiedot, kuten kunkin transientin nimi, kuormitusalue, kuormatyyppi, lähdedokumentti, esiintymisten lukumäärä sekä transientin funktio ja tähän liittyvä numeerinen data. Transientti saattaa kuvata esimerkiksi lämpötilaa ajan funktiona. Tällöin numeerinen data ilmaisee lämpötilaa tietyllä hetkellä.

Lisäksi jokaista transienttia kohden on erilaista oheistietoa, kuten esimerkiksi transientin tila ja käyttäjän syöttämä huomautus. Tilalla kuvataan sitä, onko transientti esimerkiksi syötetty osittain, syötetty kokonaan tai syötetty ja tarkastettu. Huomautus on käyttäjän vapaamuotoinen teksti, joka saattaa sisältää esimerkiksi tiedon siitä miksi transienttia ei vielä ole syötetty kokonaan. Tähän syynä saattaa olla vaikkapa epäselvyys lähdemateriaalissa.

3.1.2 Kuormitustietokannan rakenne

Kuormitustietokanta sisältää kymmeniä toisiinsa liittyviä tauluja. Lähes kaikki taulut ovat ajantasaisia ja käytössä, mutta jotkin näistä tauluista ovat jäänteitä aiemmista kehitysversioista. Tällaiset taulut on tarkoitus poistaa tietokannasta, kun korvaavat toimiviksi todetut uudet taulut otetaan käyttöön ja ne on todettu toimiviksi. Kuormitustietokannan tärkeimmät taulut ja niiden väliset yhteydet on esitetty liitteessä 1.

3.2 LoadsDb-ohjelmisto

3.2.1 LoadsDb-ohjelmiston tehtävä

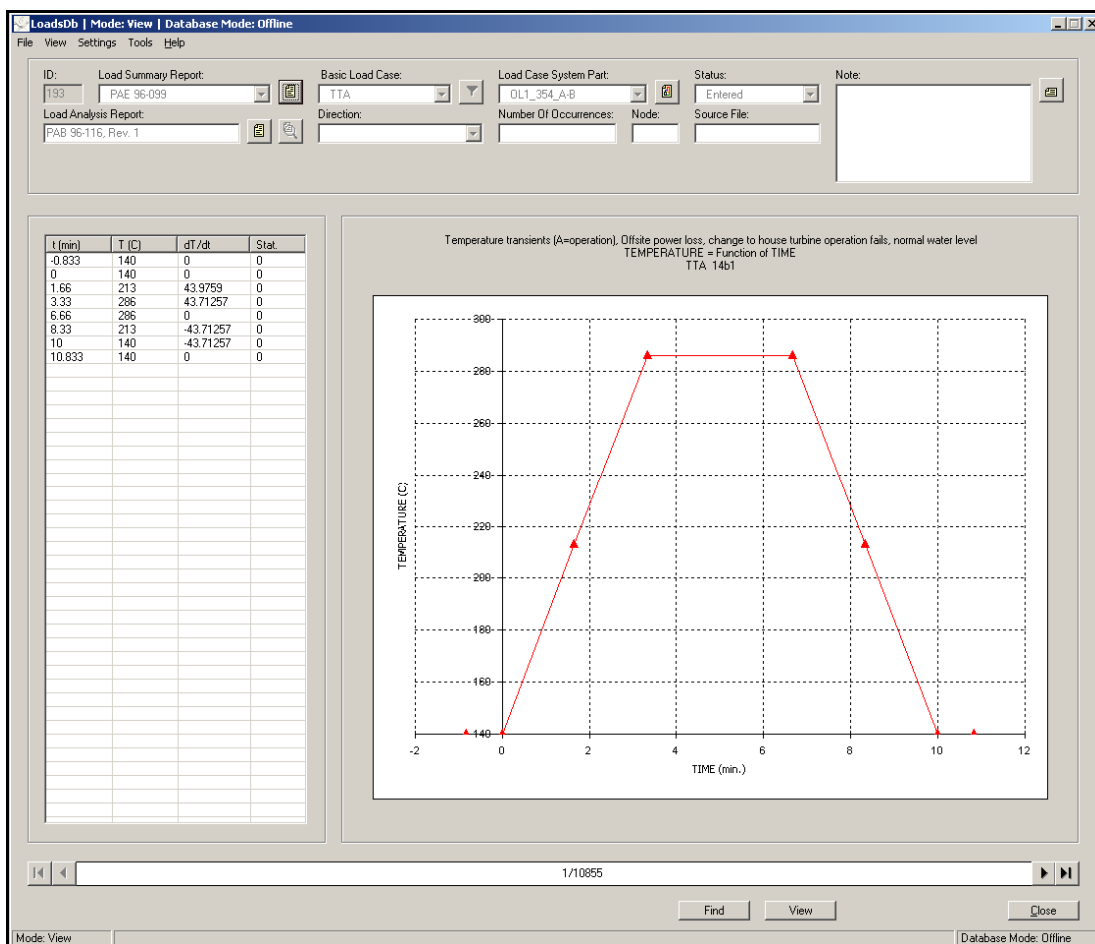
LoadsDb-ohjelmiston tarkoituksena on toimia kuormitustietokannan ylläpidon graafisena työkaluna. Lisäksi ohjelmisto sisältää kattavat ominaisuudet kuormitustietokannan sisältämän tiedon visualisointiin.

3.2.2 LoadsDb-ohjelmiston päänäkyvä

LoadsDb-ohjelmiston päänäkyvässä on mahdollista suorittaa kaikki tärkeimmät kuormitustietokantaa koskevat toimenpiteet. Päänäkyvässä on mahdollista poistaa ja muokata kuormitustietokannan sisältämiä transientteja sekä lisätä tietokantaan uusia transientteja. Lisäksi yksi päänäkyvän tärkeimmistä ominaisuuksista on syötettyjen transienttien visualisointi.

Päänäkyvässä on kaksi tilaa, katselutila ja muokkaustila. Katselutilassa käyttäjä voi ainoastaan selaila kuormitustietokannan sisältämiä tietoja, kun taas muokkaustilassa käyttäjän on mahdollista lisätä, muokata ja poistaa tietoa.

Tiedon selailun ja muokkauksen mahdollistamisen lisäksi yhtenä päänäkyvän tärkeimmistä tehtävistä on tarjota käyttäjälle pääsy ohjelmiston muihin ominaisuuksiin. Näitä ominaisuuksia ovat esimerkiksi tiedonhakulomake sekä kuormitustapausten syöttämiseen tarkoitettu ikkuna. Ohjelmiston päänäkyvä on esitetty kuvassa 5.



Kuva 5. LoadsDb-ohjelmiston päänäkymä.

4 TYÖKALUT

4.1 Sovelluskehitin

Sovelluskehittimenä käytettiin Microsoft Visual Basic 6 -kehitysympäristöä, johon oli asennettu viimeisin Service Pack 6 -päivitys. Visual Basic 6 kehitysympäristönä tarjoaa graafisen käyttöliittymän ja kattavat työkalut Windows-käyttöjärjestelmissä toimivien ohjelmistojen luomiseen. Lisäksi yksi suuri etu on kehitysympäristöön integroitu tuki tietokantayhteyksille. /3, s. 4/

Tietokantayhteys LoadsDb-ohjelmiston ja kuormitustietokannan välille muodostettiin käyttämällä ADO:a. Tämä mahdollisti sekä tiedon kyselyn että tietojen muok-

kaamisen tietokannasta sql-lausekkeilla. Lisäksi tämä kaikki saatiin aikaan vain muutamalla rivillä ohjelmakoodia.

Huonona puolena Visual Basic 6 -kehitysympäristössä on, että Microsoft on lopettanut virallisen tuen tuotteelle. Tämä saattaa tulevaisuudessa olla syy vaihtaa toiseen kehitysympäristöön. /4/

4.2 Tietokantasovellus

Kaikki tietokanta-alueen tietokannat, kuten myös kuormitustietokanta, ovat Microsoft Access -tietokantoja. Myös kuormitustietokannan jatkokehittäminen tapahtui Accessissa. Kuormitustietokannan ylläpitoon kuuluvia perustoimenpiteitä olivat esimerkiksi sekä taulujen, että niiden sisältämien tietojen muokkaus, lisäys ja poisto. Näiden perustoimenpiteiden suorittamiseen Access tarjoaa helpon ja nopean graafisen käyttöliittymän.

Edellä mainittujen perustoimenpiteiden lisäksi työssä hyödynnettiin joitakin Accessin kehittyneemmistä ominaisuuksista. Näistä ehdottomasti käyttökelpoisin oli mahdollisuus luoda tietokantakyselyjä graafisesti ja tämän jälkeen antaa Accessin muuntaa kysely sql-muotoiseksi. Tämän jälkeen oli mahdollista siirtää luotu sql-kysely sovelluskehittimen puolelle hyödynnettäväksi ohjelmakoodissa. /5, s. 357/

4.3 Piirto-ohjelma

Piirto-ohjelmaa käytettiin LoadsDb-ohjelmiston kuvakkeiden luontiin. Koska kuvakkeet olivat poikkeuksetta melko pieniä ja yksinkertaisia, ei myöskään piirto-ohjelmalta vaadittu minkäänlaisia erityisominaisuuksia. Tämän vuoksi päädyttiin käyttämään lähes jokaisen Windows-käyttöjärjestelmän mukana toimitettavaa Microsoft Paint -nimistä piirto-ohjelmaa.

4.4 Muut ohjelmat

Edellä mainittujen ohjelmien lisäksi työssä hyödynnettiin useita muita ohjelmia satunnaisesti. Näitä ohjelmia olivat esimerkiksi Adobe Reader sekä lähes kaikki Microsoft Office -tuoteperheen ohjelmat aina Wordista Outlookiin.

5 SPEKTRIT JA KERROSTUMAT

5.1 Ominaisuuden tarkoitus

Tavanomaisempien transienttien ohella kuormitustietokantaan syötetään myös spektreja ja kerrostumia. Nämä poikkeavat tavallisista transienteista niin, että niillä on transienttien perustietojen lisäksi tyypilleen ominainen lisätieto, jota kuormitustietokannassa kutsutaan nimellä load type argument.

Spektrien tapauksessa tämä lisätieto on kunkin spektriin kuuluvan transientin vaimennus eli damping. Kerrostumien tapauksessa taas kyseessä on kunkin transientin sijainti eli korkeus putkessa. Kuormitustietokannan yhteydessä tästä käytetään nimeä height. Lisäksi molemmille tyypeille ominaista on, että ne muodostavat usean transientin ryhmiä. Spektrien tapauksessa useasta yksittäisestä käyrästä muodostuu spektrikuorma, kun taas kerrostumien tapauksessa kyseessä on putken eri korkeuksista

Tarkoituksena oli mahdollistaa näiden spektrien ja kerrostumien syöttäminen kuormitustietokantaan sekä laajentaa LoadsDb-ohjelmiston visualisointiominaisuuksia kattamaan kyseiset käyrätyypit.

5.2 Muutokset tietokantaan

Jotta spektreja ja kerrostumia oli mahdollista syöttää tietokantaan, tarvittiin joitakin muutoksia. Ensimmäiseksi lisättiin tauluun tblLoadDataSource uusi kenttä nimeltään

LoadTypeArgumentValue. Tähän kenttään tallennetaan käyrätyypistä huolimatta kyseisen transientin lisätieto, kuten vaikka vaimennus.

Tämän lisäksi luotiin yksi uusi taulu nimeltään tblLoadTypeArguments sekä lisättiin viittaus taulusta tblLoadData tähän uuteen tauluun. Tämän yhteyden luova kenttä on nimeltään LTA_ID. Uusi taulu ilmoittaa lisätiedon tyyppin, kuten damping tai height, sekä onko kyseessä tavallinen käyrä vai spektri tai kerrostuma (ks. liite 1).

5.3 Ominaisuuden ohjelmallinen toteutus

5.3.1 Uusi kenttä

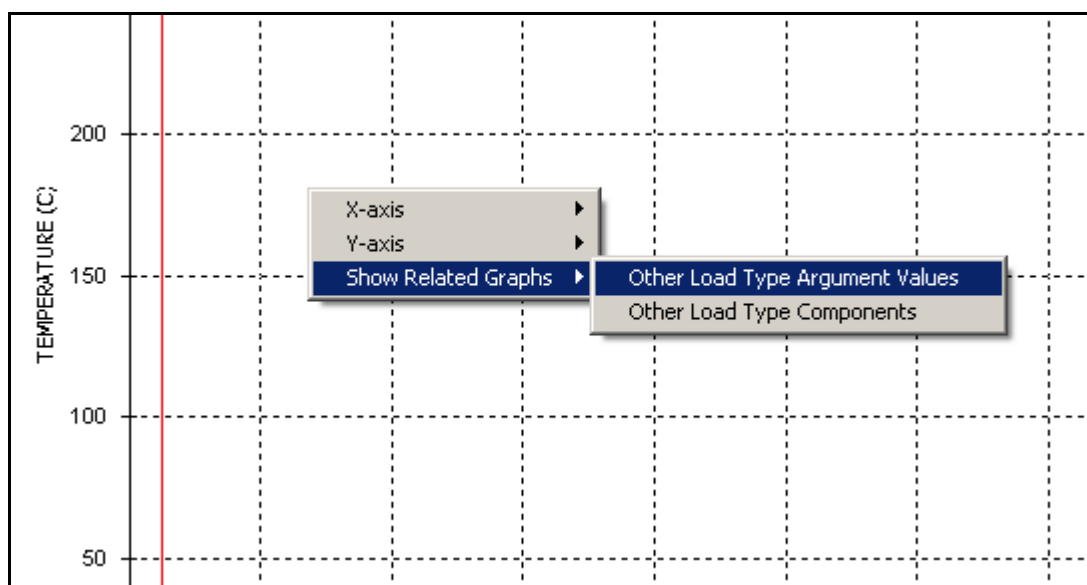
Päänäkymään lisättiin uusi kenttä tyyppikohtaisen lisätiedon syöttämistä varten. Tämä kenttä on näkyvässä ainoastaan, kun kyseessä on spektri tai kerrostuma. Kentän, sekä kentän selitteen piilottava ja näyttävä koodi on esitetty listauksessa 1.

```
Private Sub UpdateBasicValues()
... ohjelmakoodia ...
'LoadTypeArgumentName
lblLoadTypeArgVal.Caption = .Fields("LoadTypeArgumentName").Value & ":"
If strGraphType = "Slug Flow" Then
    lblLoadTypeArgVal.Visible = False
Else
    lblLoadTypeArgVal.Visible = True
End If
'LoadTypeArgumentValue
If strGraphType = "Slug Flow" Then
    cmbLoadTypeArgVal.Visible = False
Else
    If IsNull(.Fields("LoadTypeArgumentValue").Value) Or & _
        (.Fields("LoadTypeArgumentValue").Value = "") Then
        cmbLoadTypeArgVal.ListIndex = 0
    Else
        cmbLoadTypeArgVal.Text = .Fields("LoadTypeArgumentValue").Value
    End If
    cmbLoadTypeArgVal.Visible = True
End If
... ohjelmakoodia ...
End Sub
```

Listaus 1. Lisätietokentän näyttäminen ja piilotus.

5.3.2 Näytettävien käyrien lukumäärän valinta

Jotta käyttäjällä olisi mahdollisuus valita, näytetäänkö yksi vai useita käyriä, toteutettiin päänäkymään tämän mahdollistava valikko. Oletuksena valinta on pois päältä, jolloin näytetään ainoastaan tarkasteltavan transientin visualisointi. Valikon voi avata kahdella tavalla. Ensimmäinen tapa on avata valikko päänäkymän yläosassa sijaitsevasta päävalikosta. Toinen tapa on napsauttaa hiiren oikeaa painiketta käyrän kuvaajan päällä (ks. kuva 6).



Kuva 6. Näytettävien käyrien lukumäärän valinta.

5.3.3 Spektrien ja kerrostumien visualisointi

Spektrien ja kerrostumien visualisoinnin mahdollistamiseksi piti aliohjelmaa UpdateChart kehittää. Ensimmäiseksi aliohjelmaan lisättiin ohjelmakoodia, joka tarkastaa, onko käyttäjä valinnut näytettäväksi tarkasteltavaan käyrään liittyvät käyrät. Tässä tapauksessa aliohjelma yrittää etsiä muita, tarkasteltavan transientin kanssa samaan ryhmään kuuluvia käyriä. Mikäli niitä löytyy, näytetään myös nämä käyrät tarkasteltavan transientin käyrän lisäksi. Aliohjelman ohjelmakoodia on esitetty listauksessa 2. Lisäksi esimerkki kerrostumien visualisoinnista on esitetty kuvissa 7 ja 8.

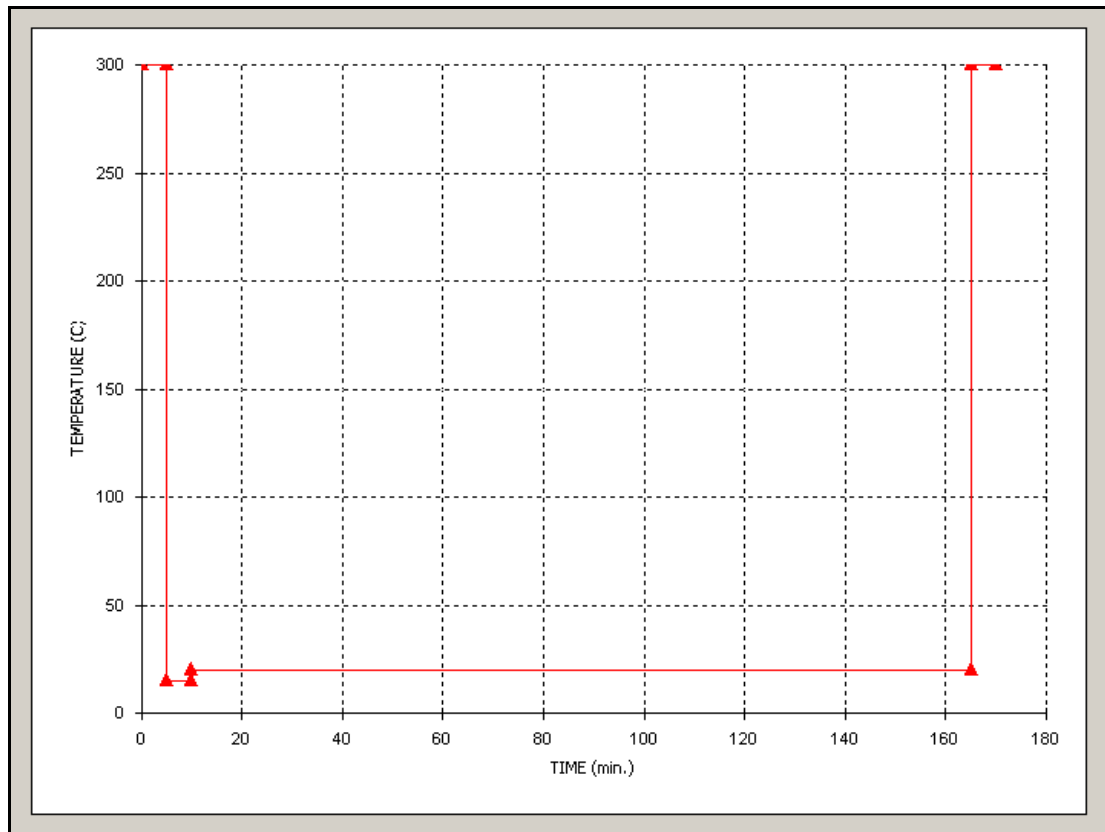
```

Private Sub UpdateChart()
... ohjelmakoodia ...
'Find transients with same LoadTypeComponent and different LoadTypeArgumentValue
'(e.g. Spectra and stratified flows)
If mnuOtherLTArgVal.Checked = True Then
    strQuerySecGraphs = "SELECT tblLoadDataSource.ID, " & _
        "tblLoadData.LoadTypeComponent FROM tblLoadDataSource " & _
        INNER JOIN tblLoadData ON tblLoadDataSource.BLC_ID = " & _
        tblLoadData.BLC_ID WHERE tblLoadData.TID_ID=" & _
        cmbBLC_ID.List(cmbBLC_ID.ListIndex, 2) & " AND " & _
        tblLoadDataSource.LoadCaseSP_ID=" & cmbLoadCaseSP.Value & _
        " AND tblLoadData.LoadTypeComponent=" & _
        cmbBLC_ID.List(cmbBLC_ID.ListIndex, 3) & _
        " AND tblLoadDataSource.ID<>" & txtID.Text & ";"

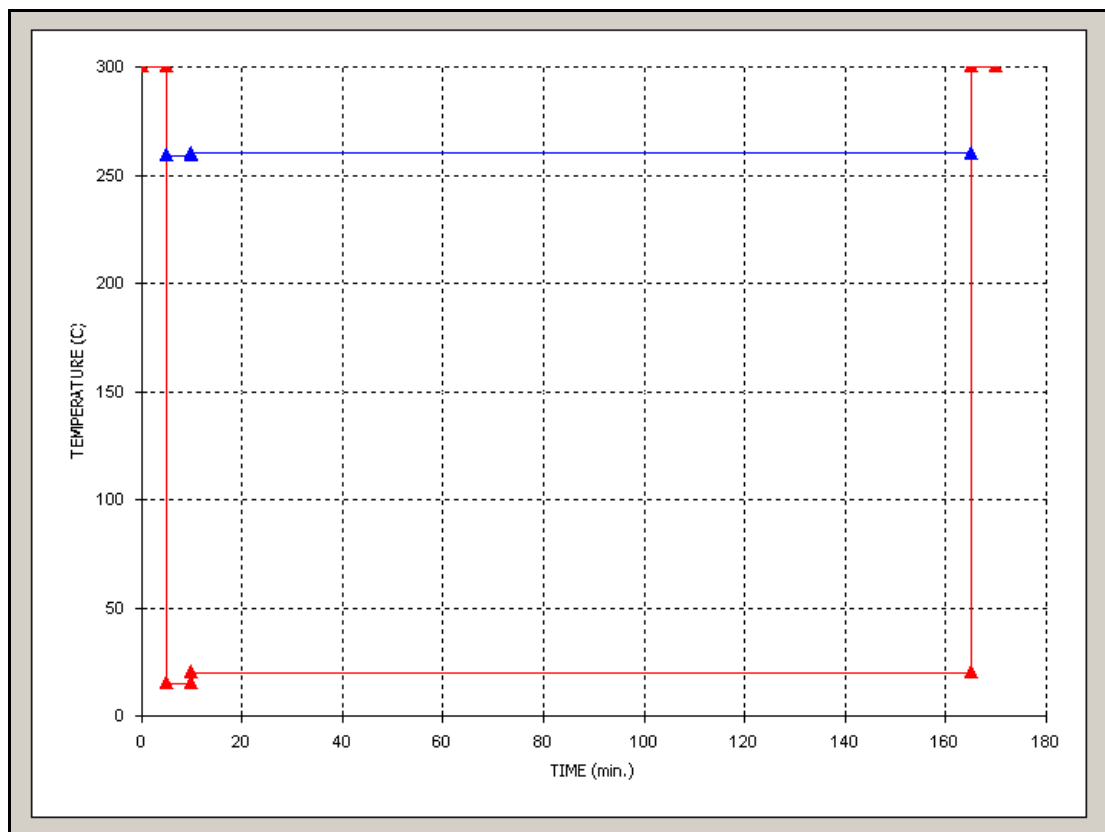
    With rs_SecGraphs
        .Open strQuerySecGraphs, pConn, adOpenStatic, adLockOptimistic
        Do Until .EOF
            clnSecGraphIDs.Add .Fields("ID").Value
            clnSecGraphLoadTypes.Add .Fields("LoadTypeComponent").Value
            .MoveNext
        Loop
        .Close
    End With
End If
... ohjelmakoodia ...
If clnSecGraphLoadTypes.Item(i) = strLoadType Then
'Load Type matches main graph -> stratification or spectra = Blue
clnChartLegendID.Add clnSecGraphIDs.Item(i)
clnChartLegendText.Add clnSecGraphLoadTypes.Item(i)
clnChartLegendColor.Add &HFF0000
.Plot.SeriesCollection(j).SeriesMarker.Auto = False
.Plot.SeriesCollection(j).DataPoints(-1).Marker.Style = VtMarkerStyleFilledUpTriangle
.Plot.SeriesCollection(j).DataPoints(-1).Marker.Size = 150
.Plot.SeriesCollection(j).Pen.Width = 1
.Plot.SeriesCollection(j).DataPoints(-1).Brush.FillColor.Set 0, 0, 255
.Plot.SeriesCollection(j).DataPoints(-1).Marker.Pen.VtColor.Set 0, 0, 255
.Plot.SeriesCollection(j).DataPoints(-1).Marker.FillColor.Set 0, 0, 255
.Plot.SeriesCollection(j).SeriesMarker.Show = True
... ohjelmakoodia ...
End Sub

```

Lista 2. Spektrien ja kerrostumien visualisoinnin ohjelmakoodi.



Kuva 7. Yksittäisen käyrän visualisointi.



Kuva 8. Kerrostuman visualisointi.

6 HAKULOMAKKEEN KEHITTÄMINEN

6.1 Hakulomake

Hakulomakkeen tarkoituksena on tarjota käyttäjälle mahdollisuus suodattaa ja etsiä kuormitustietokannan sisältämää tietoa. Koska kuormitustietokanta sisältää tuhansia transientteja, voidaan hakulomaketta pitää yhtenä LoadsDb-ohjelmiston tärkeimmistä osa-alueista. Kuvassa 9 on esitetty hakulomake, kun käyttäjä on suorittanut haun.

Search Conditions

Plant: Load Summary Report: Load Case System Part: Load Abbreviation: Transient: Load Type Component: Argument:

No Of Arguments: Graph Type: Load Analysis Report: Source File: Node: Status:

Find Clear Search Conditions Clear List Found 10 results matching the search conditions.

Load Summary...	Load Case SPCode	Description	Load...	Transient	Load Type Component	Argument	No.	Graph Type	Load Analysis R...	Num.	Status	Note
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	1a	PRESSURE	TIME	1	Slug Flow	SDC 96-1095		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	1a	TEMPERATURE	TIME	1	Slug Flow	SDC 96-1095		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	1b	PRESSURE	TIME	1	Slug Flow	SDC 96-1096		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	1b	TEMPERATURE	TIME	1	Slug Flow	SDC 96-1096		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	3	TEMPERATURE	TIME	1	Slug Flow	SDC 96-1095		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	312_2b_closed	TEMPERATURE	TIME	1	Slug Flow	SDC 96-1095		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	312_2b_lineasp...	TEMPERATURE	TIME	1	Slug Flow	SDC 96-1095		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	312_2c_closed	TEMPERATURE	TIME	1	Slug Flow	SDC 96-1095		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	312_2c_lineasp...	TEMPERATURE	TIME	1	Slug Flow	SDC 96-1095		Entered	
PAB 96-125	OL1_312_A-B	312V3 -> V1 for OL1	TTA	6p	TEMPERATURE	TIME	1	Slug Flow	SDC 96-1095		Entered	

OK Unlock/Stepping Cancel

Kuva 9. Hakulomake.

6.2 Hakukenttien suodatus

Ensimmäinen uudistus oli lisätä jokaiseen hakukenttään toiminta, jonka avulla hakukenttään asetettu valinta suodattaa kyseistä hakukenttää seuraavana olevan hakukenttän valintoja. Tämä helpottaa käyttäjää löytämään etsimänsä tiedon nopeammin sekä vähentää tuloksettomien hakujen määrää.

6.3 Uusi hakukenttä: Plant

Käyttäjäpalautteen perusteella päätettiin lisätä hakulomakkeen ensimmäiseksi hakukentäksi plant-niminen kenttä. Näin käyttäjä voi heti aluksi rajata hakunsa tietyn laitoksen transientteihin. Tämä on erittäin hyödyllistä, mikäli käyttäjä tietää transientin laitoksen, koska valinta rajaa pois mahdollisesti jopa tuhansia tuloksia.

Lisäksi edellä esitellyn ominaisuuden mukaisesti, myös plant-hakukenttä suodattaa seuraavaa hakukenttää. Koska plant-hakukenttä on hakulomakkeen ensimmäinen hakukenttä, on kyseessä koko hakulomakkeen tärkein yksittäinen suodatus. Tämä johtuu siitä, että laitoksen valinnan jälkeen kaikki muut tiedot on rajattu vain valittuun laitokseen kuuluvien joukkoon. Listauksessa 3 on esitetty laitoksen valinnan aiheuttama seuraavan hakukentän suodatus.

6.4 Muut muutokset

Edellä mainittujen muutosten lisäksi hakulomakkeeseen tehtiin joitakin pienempiä parannuksia. Tällaisia parannuksia olivat esimerkiksi käyttöliittymän selkeyttäminen sekä hakulomakkeen avautumisen nopeuttaminen. Lisäksi eräs tärkeä hakulomakkeeseen lisätty ominaisuus oli päänäkymän askelluksen lukitseminen hakutuloksiin. Tästä ominaisuudesta kerrotaan kuitenkin enemmän seuraavassa luvussa.


```

Private Sub Filter_cmbFindLoadSummaryRprt()
... ohjelmakoodia ...
With cmbFindLoadSummaryRprt
.Clear
.AddItem ""
End With

If cmbFindPlant.Text = "" Then
For i = 1 To UBound(LoadZoneReportCache)
'0=ID, 1=Tunnus, 2=revisio, 3=company, 5=revdate, 6=title
With cmbFindLoadSummaryRprt
.AddItem LoadZoneReportCache(i, 1)
.ItemData(.ListCount - 1) = LoadZoneReportCache(i, 0)
End With
Next i
Else
With clnReportIDs
Do Until .Count = 0
.Remove (.Count)
Loop
End With

For i = 1 To UBound(LoadZoneCache)
'2=Laitos, 7=Report_id
If LoadZoneCache(i, 2) = cmbFindPlant.Text Then
clnReportIDs.Add LoadZoneCache(i, 7)
End If
Next i

For i = 1 To UBound(LoadZoneReportCache)
'0=ID, 1=Tunnus, 2=revisio, 3=company, 5=revdate, 6=title
With cmbFindLoadSummaryRprt
For j = 1 To clnReportIDs.Count
If LoadZoneReportCache(i, 0) = clnReportIDs.Item(j) Then
.AddItem LoadZoneReportCache(i, 1)
.ItemData(.ListCount - 1) = LoadZoneReportCache(i, 0)
Exit For
End If
Next j
End With
Next i
End If
cmbFindLoadSummaryRprt.ListIndex = 0
End Sub

```

Listaus 3. Seuraavan hakukentän suodatus laitousyksikön valinnan jälkeen.

7 HAKUTULOSTEN ASKELLUS

7.1 Ominaisuuden tarkoitus

Käyttäjän tehdessä hakuja kuormitustietokannasta vastauksia tulee usein enemmän kuin yksi. Usein käyttäjä haluaisi tutkia näitä hakutuloksia peräkkäin. LoadsDb-ohjelmiston päänäkymän painikkeilla käyttäjän on mahdollista askeltaa transienttien välillä eteenpäin ja taaksepäin. Tiedot eivät kuitenkaan välttämättä sijaitse kuormitustietokannassa peräkkäin, joten tällä tavalla askeltamalla käyttäjän on mahdotonta tutkia suorittamansa haun hakutuloksia peräkkäin. Ainoaksi vaihtoehdoksi jää palata takaisin hakulomakkeeseen ja valita tuloksista seuraava tarkasteluun.

Koska tällainen menettely on kuitenkin kankea ja aikaa vievä, päädyttiin toteuttamaan ominaisuus, joka lukitsee askelluksen hakutuloksiin. Tällä tarkoitetaan sitä, että käyttäjän suorittaessa haun, päänäkymän painikkeet askeltavat ainoastaan hakutuloksia, kunnes käyttäjä jälleen vapauttaa askelluksen.

7.2 Ominaisuuden toteutus

7.2.1 Yleistä ominaisuuden toteutuksesta

Koska LoadsDb-ohjelmisto sisälsi jo entuudestaan hakulomakkeen sekä päänäkymän askelluspainikkeet, oli kyseiset ominaisuudet lisättävä jo olemassa olevan ohjelmakoodin joukkoon. Tämä taas aiheutti sen, että paikoitellen piti myös jo olemassa olevaa ohjelmakoodia muuttaa vastaamaan uusien ominaisuuksien vaatimuksia.

7.2.2 Hakutulosten tallennus

Jotta hakutuloksia olisi mahdollista askeltaa, toteutettiin SearchedIDs niminen luokkamoduuli, jonne hakutulokset tallennettiin. Tietojen tallennukseen käytettiin luok-

kamoduulin sisälle luotua kokoelmaa nimeltään cInIDs ja kokoelmassa liikkumiseen taas kokonaisluku muuttujaa i.

Näiden objektien avulla luokkamoduuliin toteutettiin tarvittavat ominaisuudet kokoelman hallintaan. Näitä ominaisuuksia olivat esimerkiksi kokoelman alustus, kokoelman tyhjennys, tietojen lisääminen kokoelmaan, tietojen hakeminen kokoelmasta sekä kokoelmassa liikkuminen.

7.2.3 Askelluspainikkeet

Koska askelluspainikkeet olivat olemassa jo entuudestaan, piti niiden toimintaa laajentaa ottamaan huomioon haluttiinko selata koko kuormitustietokantaa vai ainoastaan hakutuloksia. Jokaisen askelluspainikkeen napsautus kutsuu aliohjelmaa ShiftRecord.

ShiftRecord-aliohjelman toiminnallisuutta laajennettiin niin, että mikäli askellus on lukittu, askelletaankin koko tietokannan tietueiden sijaan ainoastaan hakutuloksia. Tämän jälkeen askelletaan koko tietokannan tietueissa hakutulosten osoittamaa tietuetta vastaavan tietueen kohdalle. Listauksessa 4 on esitetty aliohjelman ShiftRecord sisältämä ohjelmakoodi.

```

Private Sub ShiftRecord(ByVal strShiftPos As String)
  If strStepMode = "Normal" Then
    If strShiftPos = "First" Then
      LDSIDs.MoveFirst
    ElseIf strShiftPos = "Prev" Then
      LDSIDs.MovePrevious
    ElseIf strShiftPos = "Next" Then
      LDSIDs.MoveNext
    ElseIf strShiftPos = "Last" Then
      LDSIDs.MoveLast
    End If
  ElseIf strStepMode = "Locked" Then
    If strShiftPos = "First" Then
      SearchIDs.MoveFirst
      LDSIDs.MoveToID CLng(SearchIDs.GetID)
    ElseIf strShiftPos = "Prev" Then
      SearchIDs.MovePrevious
      LDSIDs.MoveToID CLng(SearchIDs.GetID)
    ElseIf strShiftPos = "Next" Then
      SearchIDs.MoveNext
      LDSIDs.MoveToID CLng(SearchIDs.GetID)
    ElseIf strShiftPos = "Last" Then
      SearchIDs.MoveLast
      LDSIDs.MoveToID CLng(SearchIDs.GetID)
    End If
  End If
  ... ohjelmakoodia ...
End Sub

```

Lista 4. Aliohjelman ShiftRecord ohjelmakoodi.

7.2.4 Askelluksen lukitus

Hakulomake suljetaan, kun käyttäjä on valinnut jonkin hakutuloksista ja painaa hakulomakkeen Ok-painiketta siirtyäkseen tarkastelemaan valitsemaansa hakutulosta. Tällöin hakutulokset tallennetaan kokoelmaan, josta niitä askelluksen yhteydessä selataan. Nämä toimenpiteet on esitetty listauksessa 5.

```

Private Sub cmdOk_Click()
  ... ohjelmakoodia ...
  frmParent.SearchIDs.Clear
  For i = 1 To clnSearchResultIDs.Count
    frmParent.SearchIDs.Add clnSearchResultIDs.Item(i)
  Next i
  ChangeRecord
  strExitState = "Ok"
  Unload Me
End If
End Sub

```

Lista 5. Hakulomakkeen sulkeminen Ok-painikkeella.

Kun hakulomake suljetaan, palataan takaisin päänäkymään. Tällöin tarkistetaan, poistuttiinko hakulomakkeelta Ok-painikkeen kautta. Mikäli poistuttiin, niin askellus lukitaan ja käyttäjän valitsema hakutulos näytetään käyttäjälle. Edellä mainitut toimenpiteet suorittava ohjelmakoodi on listauksessa 6.

```
Private Sub cmdFind_Click()
... ohjelmakoodia ...
'check how the form was closed
If frmFind.strExitState = "Ok" Then
    ChangeStepMode "Locked"
    If ((LDSIDs.MoveToID(frmFind.intLoadDataSourceID)) And & _
(SearhIDs.MoveToID(frmFind.intLoadDataSourceID))) Then
        UpdateData
... ohjelmakoodia ...
    End If
End Sub
```

Listaus 6. Askelluksen lukitus käyttäjän sulkiessa hakulomakkeen.

7.2.5 Askelluksen vapautus

7.2.5.1 Yleistä

Käyttäjällä pitää olla myös mahdollisuus palauttaa askellus tilaan, jossa ohjelmisto askeltaa kaikkia kuormitustietokannan tuloksia. Tämän käyttäjä voi tehdä sekä päänäkymästä, että hakulomakkeesta.

7.2.5.2 Hakulomake

Hakulomakkeelle toteutettiin komentopainike, jota painettaessa hakulomake suljetaan ja päänäkymään palattaessa askellus vapautuu. Hakulomakkeen osuus on esitetty listauksessa 7 ja päänäkymässä suoritettava koodi listauksessa 8.

```
Private Sub cmdUnlockStepping_Click()
    strExitState = "Unlock"
    Unload Me
End Sub
```

Listaus 7. Komentopainikkeen cmdUnlockStepping toiminta.

```

Private Sub cmdFind_Click()
... ohjelmakoodia ...
'check how the form was closed
... ohjelmakoodia ...
ElseIf frmFind.strExitState = "Unlock" Then
    ChangeStepMode "Normal"
End If
End Sub

```

Listaus 8. Askelluksen vapautus käyttäjän sulkiessa hakulomakkeen.

Lisäksi hakulomakkeen lataukseen lisättiin ohjelmakoodi, joka tarkistaa, onko askellus lukittu. Mikäli hakua ei ole lukittu, otetaan askelluksen vapauttava komentopainike pois käytöstä. Hakulomakkeen latautuessa suoritettava ohjelmakoodi on esitetty listauksessa 9.

```

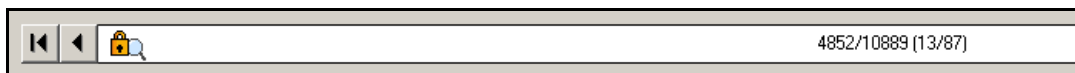
Private Sub Form_Load()
... ohjelmakoodia ...
'Enable/Disable Unlock Stepping button
If frmParent.strStepMode = "Locked" Then
    cmdUnlockStepping.Enabled = True
Else
    cmdUnlockStepping.Enabled = False
End If
... ohjelmakoodia ...
End Sub

```

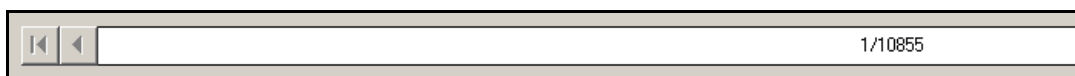
Listaus 9. Askelluksen lukituksen tarkistus.

7.2.5.3 Päänäkymä

Vastaavasti päänäkymässä askelluksen vapautus tapahtuu painamalla askelluspainikkeiden läheisyydessä sijaitsevaa lukon kuvaketta (ks. kuvat 10 ja 11). Listauksessa 10 on esitetty kuvakkeen painalluksesta suoritettava ohjelmakoodi.



Kuva 10. Askellus lukittu hakutuloksiin.



Kuva 11. Vapaa askellus.

```
Private Sub imgStepLock_Click(Index As Integer)
    ChangeStepMode "Normal"
End Sub
```

Listaus 10. Kuvakkeen imgStepLock toiminta.

7.2.6 Toimenpiteet askelluksen lukituksen ja vapautuksen yhteydessä

Kun askelluksen tilaa muutetaan eli askellus lukitaan tai vapautetaan, kutsutaan ChangeStepMode -nimistä aliohjelmää. Tässä aliohjelmassa suoritetaan joukko toimenpiteitä, joilla huolehditaan siitä, että jokainen asia, johon askelluksen tila vaikuttaa, on asetettu valitun askelluksen tilan edellyttämällä tavalla.

Tällaisia asioita ovat esimerkiksi tiettyjen objektien näkyvyys, muuttujien arvot sekä käyttäjälle näkyvät sanomat. Nämä toimenpiteet on esitetty listauksessa 11.

```
Private Sub ChangeStepMode(ByVal strMode As String)
    If strMode = "Normal" Then
        strStepMode = "Normal"
        lblRecord.ToolTipText = "Double-click To Jump To ID"
        lblRecord.Caption = LDSIDs.CurrentIndex & "/" & LDSIDs.Count
        SearchIDs.Clear
        'Hide Step Locked icons
        For Each img In imgStepLock
            img.Visible = False
        Next img
    ElseIf strMode = "Locked" Then
        strStepMode = "Locked"
        lblRecord.ToolTipText = ""
        'Show Step Locked icons
        For Each img In imgStepLock
            img.Visible = True
        Next img
    End If
    CheckShiftRecordButtons
End Sub
```

Listaus 11. Aliohjelman ChangeStepMode ohjelmakoodi.

8 ALUEKUVAN NÄYTTÄMINEN

8.1 Ominaisuuden tarkoitus

Jokainen kuormitustietokannan sisältämä transientti on kytketty johonkin kuormitusalueeseen. Nämä kuormitusalueet on kuvattu transientin lähdedokumentissa (ks. liite 2). Usein käyttäjälle on tarpeellista saada nopeasti esille transienttiin liittyvä aluekuva. Vaikka tämä aluekuva on mahdollista saada näkyviin avaamalla transienttiin liitetty lähdedokumentti ja etsimällä tästä dokumentista oikea sivu, on se usein hidasta.

Lisäksi joissakin tapauksissa lopullinen, transienttien syöttämiseen käytetty aluejako, saattaa poiketa alkuperäisestä, lähdedokumentin sisältämästä aluejaosta. Tällöin kyseessä on mukautettu aluejako eli niin kutsuttu ”TVO aluejako”, joka on kuvattu omassa dokumentissaan. Tällaisissa tapauksissa transientin lähdedokumentin ja todellisuudessa käytetyn aluejaon välinen ero saattaa sekoittaa käyttäjää.

8.2 Ominaisuuden toteutus

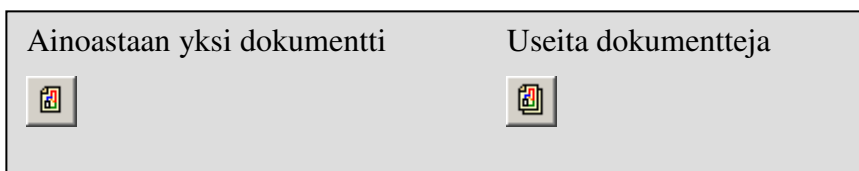
8.2.1 Yleistä ominaisuuden toteutuksesta

Edellä mainittujen epäkohtien vuoksi päädyttiin toteuttamaan ohjelmiston päänäkömään komentopainike, jonka kautta lähdedokumentti aukeaa välittömästi oikealle, aluekuvan sisältävälle sivulle. Useimmissa tapauksissa sähköinen lähdedokumentti on pdf-muotoinen tiedosto. Koska ainoastaan joissakin poikkeustapauksissa lähdedokumentti saattaa olla myös esimerkiksi Word- tai Excel-dokumentti, päätettiin komentopainikkeen tukevan ainoastaan pdf-muotoisia dokumentteja.

Mikäli tällaisia dokumentteja ei löydy, asetetaan komentopainike pois käytöstä. Kysymyksessä ei kuitenkaan ole vakava puute, koska tällaisia tapauksia on erittäin vähän ja näissäkin poikkeustapauksissa käyttäjän on mahdollista avata dokumentti toista kautta.

8.2.2 Saatavilla olevien dokumenttien lukumäärän tarkastus

Eräs tärkeä ominaisuus on myös se, että käyttäjän on mahdollista nähdä yhdellä vilkaisulla, onko saatavilla useita dokumentteja. Tämän saavuttamiseksi komentopainikkeen kuvake muuttuu, mikäli sekä alkuperäinen lähdedokumentti, että mukautetun aluejaon sisältävä dokumentti ovat olemassa (ks. kuva 12).



Kuva 12. Komentopainikkeen vaihtuvat kuvakkeet.

Saatavilla olevien dokumenttien määrä tarkastetaan aina transientin tietoja ladattaessa. Tämä toteutettiin lisäämällä hieman ohjelmakoodia aliohjelmaan `UpdateBasicValues`. Lisätty ohjelmakoodi ilmenee listauksista 12 ja 13.

Tämä ohjelmakoodi tarkistaa ensin, onko kyseiseen transienttiin liitetty dokumentti pdf-muotoinen. Mikäli näin ei ole, tarkistetaan löytyykö tietokannasta edellä mainittuun dokumenttiin liitettyjä pdf-muotoisia dokumentteja. Tämän jälkeen tarkistetaan vielä, löytyykö mukautetun aluejaon sisältäviä dokumentteja. Kun saatavilla olevien dokumenttien määrä ja tyyppi on selvillä, asetetaan kaikki tarvittavat muuttujat ja objektit tilanteen edellyttämään tilaan.

```

Private Sub UpdateBasicValues()
... ohjelmakoodia ...
'Check if Original Loading Zone report exists (.pdf)
'Check if the selected main document is in the .pdf form
strQueryLZ = "SELECT LinkDocumentsTbl.Report_ID FROM LinkDocumentsTbl " & _
"WHERE ((LinkDocumentsTbl.Report_ID=" & intLoadSummaryRpt & ") & _
" AND (LinkDocumentsTbl.LinkDescription='Main Document') AND " & _
(LinkDocumentsTbl.LinkPath LIKE "%\\Keskus1%.pdf%");"

With rs_lz
.Open strQueryLZ, pConnRPT, adOpenStatic, adLockOptimistic
If .BOF And .EOF Then
'Report doesn't exist
blnOrigLZExists = False
Else
'Report exists
blnOrigLZExists = True
intOrigLZReportID = intLoadSummaryRpt
End If
.Close
End With
If blnOrigLZExists = False Then
'Check if there are other main documents, that are in the .pdf form
strQueryLZ = "SELECT ReportsTbl.Report_ID, LiitteetTbl.Reference_ID, " & _
"ReportsTbl_1.Report_ID, ReportsTbl_1.DocumentSpecific " & _
"FROM ((ReportsTbl INNER JOIN LiitteetTbl ON ReportsTbl.Report_ID " & _
"= LiitteetTbl.Report_ID) INNER JOIN ReportsTbl AS ReportsTbl_1 ON " & _
"LiitteetTbl.Reference_ID = ReportsTbl_1.Report_ID) INNER JOIN " & _
"LinkDocumentsTbl ON ReportsTbl_1.Report_ID = LinkDocumentsTbl.Report_ID " & _
"WHERE (((ReportsTbl.Report_ID)=" & intLoadSummaryRpt & ") AND " & _
"((ReportsTbl_1.DocumentSpecific)<>'TVO Loading Zone') AND " & _
"(LinkDocumentsTbl.LinkDescription='Main Document') AND " & _
"(LinkDocumentsTbl.LinkPath LIKE "%\\Keskus1%.pdf%");"

With rs_lz
.Open strQueryLZ, pConnRPT, adOpenStatic, adLockOptimistic
If (.BOF And .EOF) Then
'Report doesn't exist
blnOrigLZExists = False
Else
'Report exists
blnOrigLZExists = True
intOrigLZReportID = .Fields("ReportsTbl_1.Report_ID").Value
End If
.Close
End With
End If

```

Listaus 12. Aliohjelmaan UpdateBasicValues lisätty ohjelmakoodi.

```

'Check if TVO Loading Zone report exists
strQueryLZ = "SELECT ReportsTbl.Report_ID, LiitteetTbl.Reference_ID, " & _
"ReportsTbl_1.Report_ID, ReportsTbl_1.DocumentSpecific " & _
"FROM ((ReportsTbl INNER JOIN LiitteetTbl ON ReportsTbl.Report_ID = " & _
"LiitteetTbl.Report_ID) INNER JOIN ReportsTbl AS ReportsTbl_1 ON " & _
"LiitteetTbl.Reference_ID = ReportsTbl_1.Report_ID) INNER JOIN " & _
"LinkDocumentsTbl ON ReportsTbl_1.Report_ID = LinkDocumentsTbl.Report_ID " & _
"WHERE (((ReportsTbl.Report_ID)=" & intLoadSummaryRpt & ") AND " & _
"((ReportsTbl_1.DocumentSpecific)='TVO Loading Zone') AND " & _
"(LinkDocumentsTbl.LinkDescription='Main Document') AND " & _
"(LinkDocumentsTbl.LinkPath LIKE '%\\Keskus1%.pdf%'));"

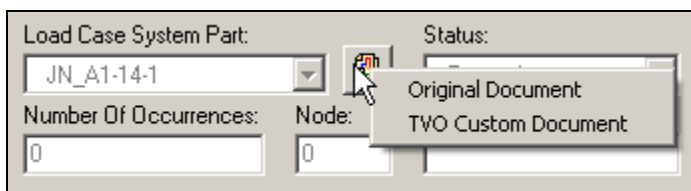
With rs_Lz
.Open strQueryLZ, pConnRPT, adOpenStatic, adLockOptimistic
If (.BOF And .EOF) Then
'Report doesn't exist
blnTVOLZExists = False
cmdShowLoadingZone.Picture = imlButtons.ListImages("imgLZ").Picture
If blnOrigLZExists Then
intNumOfReports = 1
cmdShowLoadingZone.Enabled = True
Else
cmdShowLoadingZone.Enabled = False
intNumOfReports = 0
End If
Else
'Report exists
blnTVOLZExists = True
intTVOLZRptID = .Fields("ReportsTbl_1.Report_ID").Value
If blnOrigLZExists Then
cmdShowLoadingZone.Picture = imlButtons.ListImages("imgLZMulti").Picture
intNumOfReports = 2
Else
cmdShowLoadingZone.Picture = imlButtons.ListImages("imgLZ").Picture
intNumOfReports = 1
End If
End If
.Close
End With
End Sub

```

Listaus 13. Aliohjelmaan UpdateBasicValues lisätty ohjelmakoodi.

8.2.3 Komentopainikkeen toteutus

Komentopainikkeella cmdShowLoadingZone on kaksi toimintoa. Käyttäjän napsauttaessa komentopainiketta tarkastetaan, kumpi toiminnoista suoritetaan. Mikäli saatavilla on ainoastaan yksi dokumentti, avataan kyseinen dokumentti käyttäjälle. Jos taas dokumentteja on useita, avataan käyttäjälle lista, josta käyttäjä voi valita, haluaako tarkastella alkuperäisen aluejaon vai mukautetun aluejaon sisältävää dokumenttia (ks. kuva 13 ja listaus 14).



Kuva 13. Lista dokumenteista.

```

Private Sub cmdShowLoadingZone_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
  If Button = 1 Then
    If intNumOfReports = 1 Then
      If blnTVOLZExists Then
        ShowLoadingZone intTVOLZRptID
      Else
        ShowLoadingZone cmbLoadSummaryRPT.Value
      End If
    ElseIf intNumOfReports = 2 Then
      PopupMenu mnuPopUpLoadZone
    End If
  End If
End Sub

```

Listaus 14. Painikkeen cmdShowLoadingZone toiminta nostettaessa hiiren painike ylös.

8.2.4 Dokumentin avaaminen

Dokumenttia avattaessa haetaan tietokannasta ensin dokumentin sijainti. Tämän jälkeen haetaan vielä aluekuvan sisältävän sivun numero. Mikäli sivunumeroa ei syystä tai toisesta löydy, asetetaan dokumentti aukeamaan ensimmäiseltä sivulta. Lopuksi avataan dokumentti valitulta sivulta käyttäen Shell-funktiota. Tämä kaikki tehdään aliohjelmassa ShowLoadingZone, jonka ohjelmakoodi on esitetty listauksessa 15.

```

Private Sub ShowLoadingZone(ByVal intReportID As Integer)
... ohjelmakoodia ...
'Retrieve document's path
strQueryLZ = "SELECT LinkDocumentsTbl.LinkPath FROM LinkDocumentsTbl " & _
"WHERE ((LinkDocumentsTbl.Report_ID=" & intReportID & ") AND " & _
"(LinkDocumentsTbl.LinkDescription='Main Document') AND " & _
"(LinkDocumentsTbl.LinkPath LIKE '%\\Keskus1%.pdf%'));"

With rs_lz
.Open strQueryLZ, pConnRPT, adOpenStatic, adLockOptimistic
If .BOF And .EOF Then
.MsgBox "Link to the document is invalid.", vbExclamation + vbOKOnly, "LoadsDb"
.Close
.Exit Sub
Else
.strLinkPath = .Fields("LinkPath").Value
.End If
.Close
End With

strLinkPath = Replace(strLinkPath, "#", "")

'Retrieve document's page number
strQueryLZ = "SELECT SpecialValuesTbl.SpecialValue " & _
"FROM LinkDocumentsTbl INNER JOIN (SpecialValuesTbl INNER JOIN " & _
"SpecialFeaturesTbl ON SpecialValuesTbl.SpecialValue_ID = " & _
"SpecialFeaturesTbl.SpecialValue_ID) ON LinkDocumentsTbl.Report_ID = " & _
"SpecialFeaturesTbl.Report_ID WHERE ((LinkDocumentsTbl.Report_ID=" & _
intReportID & ") AND (SpecialFeaturesTbl.SpecialName_ID=2) AND " & _
"(LinkDocumentsTbl.LinkDescription='Main Document') AND " & _
"(LinkDocumentsTbl.LinkPath LIKE '\\%.pdf%'));"

With rs_lz
.Open strQueryLZ, pConnRPT, adOpenStatic, adLockOptimistic
If .BOF And .EOF Then
.intPageNumber = 1
Else
.intPageNumber = .Fields("SpecialValue").Value
.End If
.Close
End With

On Error GoTo errShell
strShell = gAcrobatReaderPath & " /a ""page=" & intPageNumber & _
" & pagemode=none"" "" & strLinkPath & """"

Shell strShell, vbMaximizedFocus

.Exit Sub
errShell:
msgbResult = MsgBox("Couldn't find Acrobat/Reader executable." & vbCrLf & _
"Would you like to configure the path now?", vbExclamation + vbYesNo, "LoadsDb")
If msgbResult = vbYes Then
.Call mnuPreferences_Click
.End If
End Sub

```

Listaus 15. Aliohjelman ShowLoadingZone ohjelmakoodi.

9 YHTEENVETO

Työn alkaessa tavoitteena oli toteuttaa kuusi ominaisuutta LoadsDb-ohjelmistoon sekä tehdä näiden ominaisuuksien vaatimat muutokset kuormitustietokantaan. Jo ennen työn aloittamista oli ennakoitu, että tavoitetta saatettaisiin joutua rajaamaan uudelleen. Opinnäytetyön aihe päätettiin kuitenkin asettaa varmasti riittävän laajaksi, koska aiheen rajaus suppeammaksi myöhemmin olisi selkeästi helpompaa, kuin aiheen myöhäinen laajentaminen.

Työn edetessä ja ominaisuuksien paisuessa huomattiinkin, että kaikkia kuutta ominaisuutta ei olisi mahdollista toteuttaa määritettynä aikana. Tästä ei kuitenkaan aiheutunut ongelmia, koska tämä oli ennakoitu jo suunnitteluvaiheessa. Opinnäytetyön aihe rajattiin uudelleen käsittämään ainoastaan neljä ominaisuutta.

Nämä neljä ominaisuutta saatiin toteutettua, ja ne ovat olleet käyttäjien testattavinaikin jo jonkin aikaa. Käyttäjäpalautte ominaisuuksista on ollut positiivista, mikä johtuu varmasti osaltaan myös siitä, että käyttäjät itse saivat palautteellaan vaikuttaa ominaisuuksiin melko paljon. Tämä työskentelytapa osoittautuikin erittäin hyväksi, koska se mahdollisti ominaisuuksien paremman käytettävyyden sekä virheiden nopean havaitsemisen ja korjaamisen.

Työmäärien aliarvioiminen tuntuu olevan ohjelmistotuotannossa enemmänkin sääntö kuin poikkeus /6, s. 25/. Koska työmäärä alussa asetettiin tietoisesti huomattavaksi, en koe epäonnistuneeni projektissa, vaikka kaikkia suunniteltuja ominaisuuksia ei ehdittykään toteuttaa, vaan olen tyytyväinen opinnäytetyöni tuloksiin.

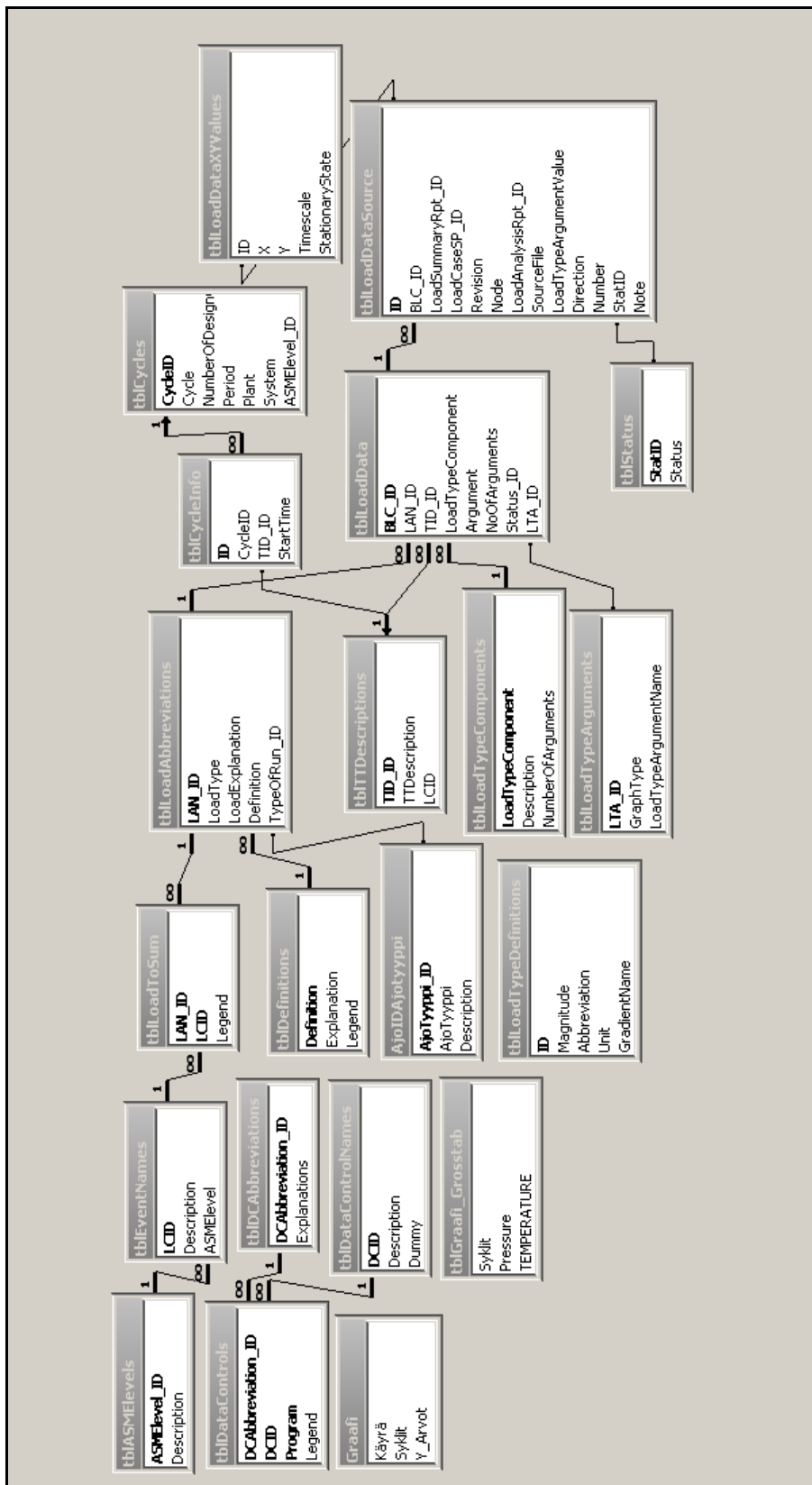
Uskon, että opinnäytetyöstä saatujen kokemusten perusteella tulevilla ohjelmistoprojekteilla työmäärän hahmottaminen ja arviointi on huomattavasti helpompaa. Lisäksi työ antoi lisäoppia lähes jokaisella ohjelmistosuunnittelun ja -tuotannon osa-alueella.

LÄHTEET

- 1: Smeekes P. & Kuuluvainen O. Opinnäytetyön aiheen tarkennus. [Kokous, 4.1.2008]
- 2: Smeekes P. The TVO Piping and component Analysis and Monitoring System (PAMS). [Word-dokumentti, viitattu 8.4.2008]
- 3: Microsoft. 2001. Visual Basic 6 – Ohjelmoijan käsikirja. Jyväskylä: ITPress
- 4: Microsoftin elinkaarituen haku. [Verkkodokumentti, viitattu 20.4.2008] Saatavissa:
<http://support.microsoft.com/lifecycle/search/?sort=PN&alpha=Visual+Basic+6&Filter=FilterNO>
- 5: Viescas J. 2000. Access 2000 käyttäjän käsikirja. Jyväskylä: ITPress
- 6: Haikala I. & Märijärvi J. 2004. Ohjelmistotuotanto. Hämeenlinna: Talentum Media Oy

LIITE 1

KUORMITUSTIETOKANNAN TAULUT JA NIIDEN VÄLISET YHTEYDET



JÄRJESTELMÄN 311 ALUEKUVA

ABB Atom AB

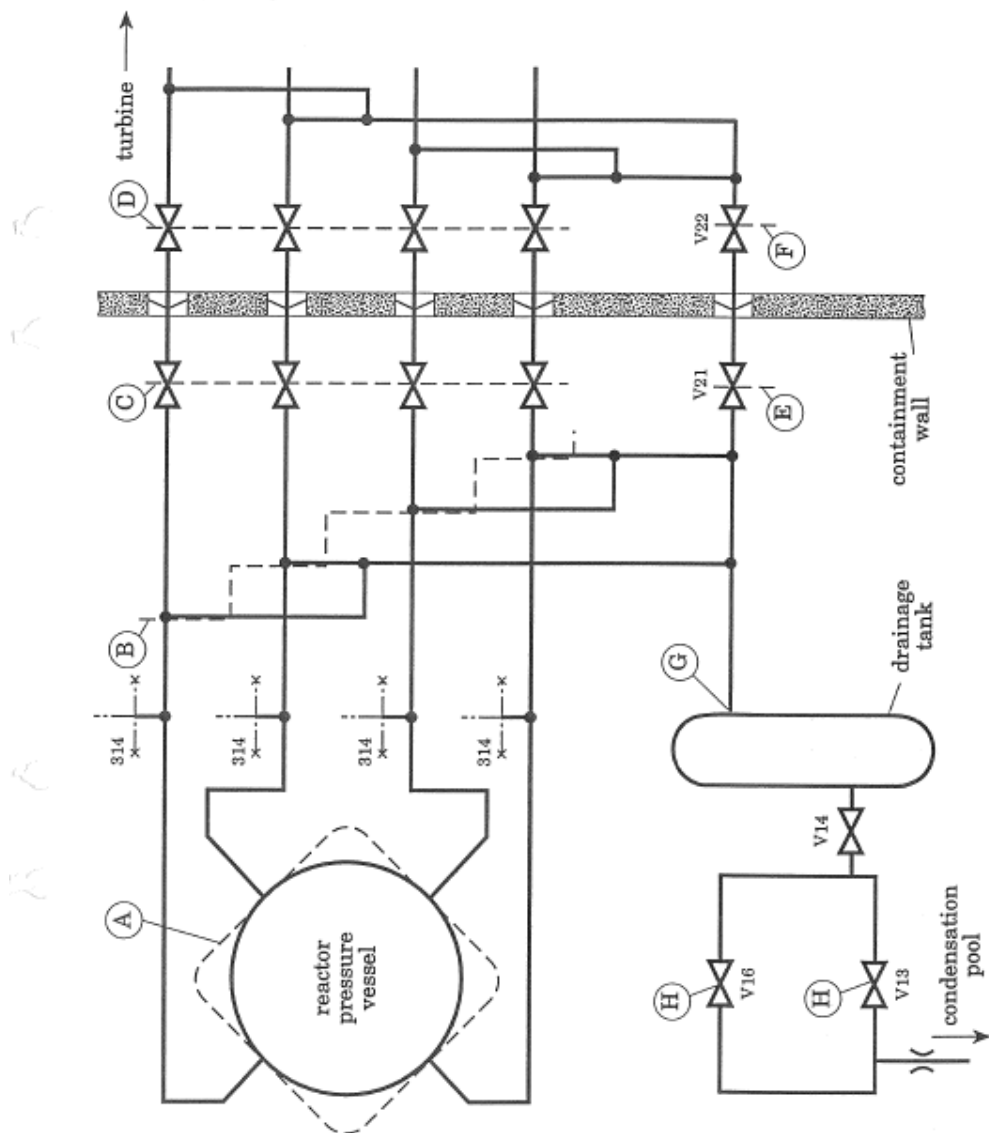
RAPPORT/REPORT
96-10-07

PAB 96-116
Sida - Page 27
Rev 1

TVO I/II - Temperature loads - Class 1 Piping systems

6 DIAGRAMS

6.1 SYSTEM 311



Denna handling får ej utan vårt medgivande kopieras. Den får ej heller delgävas annan eller viljest obehörigen användas. Övertäckelse följer beivras med stöd av gällande lag. ABB Atom

This document must not be copied without our written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose. Contravention will be prosecuted. ABB Atom

