

Opinnäytetyö (AMK)
Tietotekniikan koulutusohjelma
Ohjelmistotuotanto
2013

Vesa Niemi

KÄYTTÖLIITTYMÄN SUUNNITTELU JA TOTEUTUS



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma | Ohjelmistotuotanto

30.5.2013 | 39 sivua

Tiina Fern

Vesa Niemi

KÄYTTÖLIITTYMÄN SUUNNITTELU JA TOTEUTUS

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa toimiva ja tehokas käyttöliittymä jo olemassa olevalle ohjelmistolle. Käyttöliittymä suunniteltiin ja toteutettiin ETP (Enko Test Platform) -ohjelmistolle. Tärkeimpänä tavoitteena oli suunnitella käyttöliittymä, joka perustuu käyttöliittymien suunnitteluun kauan käytettyihin menetelmiin. Tehtävänä oli toteuttaa suunnittelun pohjalta käyttäjäystävällinen kokonaisuus käyttöliittymälle, unohtamatta nykyaikaista grafiikkaa.

Opinnäytetyö toteutettiin käyttöliittymän suunnittelun pohjalta. Suunnittelumenetelminä käytettiin jo hyväksi todettuja standardeja, esimerkiksi Jakob Nielsenin käytettävyyden säännöt käyttöliittymää suunniteltaessa. Myös rautalankamallit todettiin hyödyllisiksi käyttöliittymää luotaessa.

Työn tuloksena saatiin käyttövalmis käyttöliittymä ETP-ohjelmistoon, jonka käyttöliittymän suunnittelu ja toteutus aloitettiin täysin alusta. Suurimpana haasteena oli käyttöliittymän yhdistäminen yhdeksi toimivaksi kokonaisuudeksi. Myös aiempi graafinen toteutus päivitettiin nykystandardeihin sopivaksi.

Tämän opinnäytetyön avulla saatiin tietoa, mitä käyttöliittymän suunnittelussa tulisi oikein ottaa huomioon ja miten toteutuksessa tulisi edetä suunnittelun pohjalta. Jatkossa ETP-ohjelmisto käyttää uutta käyttöliittymää, joka pohjautuu tämän opinnäytetyön tuloksiin.

ASIASANAT:

Käyttöliittymän suunnittelu, käyttöliittymän toteutus, graafinen suunnittelu, graafinen toteutus

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Software Engineering

30 May 2013 | 39 pages

Tiina Fern

Vesa Niemi

USER INTERFACE DESIGN AND IMPLEMENTATION

The purpose of this thesis was to design and implement a functional and effective user interface for already existing software. User interface was designed and implemented for ETP (Enko Test Platform) software. The aim was to design the interface based on methods used for design for very long time. The task was to create interface by using user-friendly design base and not forgetting modern graphics.

The thesis was implemented on basis of the user interface design. Methods of the design were proven standards e.g. Jakob Nielsen's usability norms when designing user interface. Also the wireframes were useful.

The result was a ready-to-use interface for ETP software the design and implementation of which were started from the start. The major challenge was to create an interface, which works as single functional entity. Also graphical implementation was updated to fit current standards.

This thesis provides information on how the user interface design should be considered and how implementation should proceed from basis of the design. In the future ETP software uses new user interface which is based on the findings of this thesis.

KEYWORDS:

User interface design, user interface implementation, graphic design, graphic implementation

SISÄLTÖ

KÄYTETYT LYHENTEET	5
1 JOHDANTO	6
2 ENKO TEST PLATRORM (ETP) YLEISESTI	7
3 LABVIEW-OHJELMISTO	9
3.1 LabView yleisesti	9
3.2 Virtuaaliohjelmat	10
3.3 Kontrollit	10
4 KÄYTTÖLIITTYMÄN SUUNNITTELU	12
4.1 Toimiva käyttöliittymä	12
4.2 Rautalankamallit	13
4.3 Käytettävyys	14
4.4 Graafinen suunnittelu	17
4.5 Värimaailma	18
5 KÄYTTÖLIITTYMÄN TOTEUTUS	19
5.1 ETP:n ongelmat ja haasteet	19
5.2 Yleistä toteutuksesta	20
5.3 Ominaisuudet	24
5.4 Graafinen toteutus	24
5.5 Käyttöliittymän rakenne	25
5.6 Värimaailman toteutus	29
6 KÄYTTÖLIITTYMÄN TESTAUS	31
7 YHTEENVETO	36
LÄHTEET	38

KÄYTETYT LYHENTEET

ETP	yrityksen TestHouse Enko Oy ohjelmisto, Enko Test Platform
DUT	testattava laite, Device Under Test
GIMP	kuvankäsittelyohjelma, GNU Image Manipulation Program

1 JOHDANTO

Toimivan käyttöliittymän rakentaminen on nykyään olennaista niin verkossa, puhelimissa kuin työpöytäsovelluksissa. Käyttöliittymä tulisi rakentaa käyttäjälle mahdollisimman helppolukuisesti ja työhön tarkoitettuihin tehtäviin tehokkaiksi. Aina uuteen sovellukseen tutustuessa käyttäjä tulee huomioimaan käyttöliittymän ensimmäiseksi ja tämän takia siihen tulee laittaa paljon aikaa ja vaivaa, jotta sovellus on mahdollisimman helppokäyttöinen.

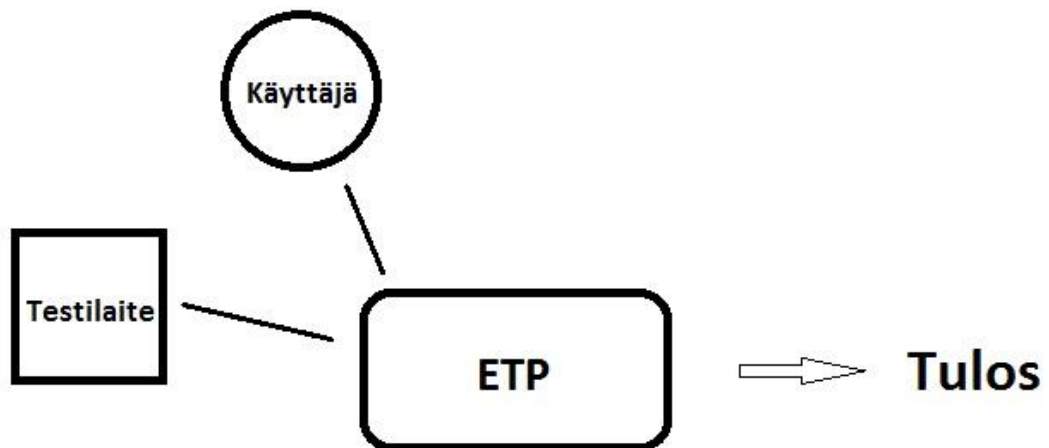
Työn tarkoituksena oli uudistaa ETP:n käyttöliittymä vastaamaan nykyajan tarpeita niin käytettävyyden kuin visuaalisuudenkin puolesta. Käyttöliittymästä tehtiin selkeä, jotta käyttäjä pärjäisi sovelluksen kanssa sulavasti niihin kuuluvien työkuvien kanssa. Tärkeät asiat tuotiin näin mahdollisimman esille ja niistä tehtiin nopeasti käytettäviä ominaisuuksia.

Opinnäytetyö koostuu käyttöliittymän suunnittelu ja teososasta, jonka sain tehdä TestHouse Enko Oy -yritykselle. Käyttöliittymä koskee testausohjelmistoa nimeltään Enko Test Platform (ETP). Ohjelmisto on rakennettu testaamaan erilaisten laitteiden dataa, joka tallennetaan erillisiin tiedostoihin jatkokäsittelyä varten. ETP:ssä on selvennetty käyttöliittymää ja tuotu erilaiset ominaisuudet nopeammin ja helpommin käyttäjän saataville.

Käyttöliittymän suunnitteluosassa on painostettu jo olemassa oleviin standardeihin, joita tulisi huomioida käyttöliittymän suunnittelussa. Näitä yleisiä standardeja tullaan käsittelemään läpikotaisesti ETP:n esimerkein. Käyttöliittymän toteutus pohjautuu vahvasti graafiseen toteutukseen, ongelmien selvittelyyn sekä ominaisuuksien katselmoineista aina testaukseen asti.

2 ENKO TEST PLATRORM (ETP) YLEISESTI

Enko Test Platform eli ETP on TestHouse Enko Oy:n kehittämä ohjelmisto, joka on luotu LabView-ohjelmistoympäristöllä. ETP on erilaisten laitteiden testausalusta, joka ajaa erillisiä testejä radiotaajuuksia käyttäville mobiilipäätelaitteille tiettyjen sekvenssien mukaisesti. Testausajon aikana ETP tallentaa tulokset erillisiin datatiedostoihin, joita voidaan jatkokäsitellä myöhemmin tarpeiden mukaisesti. ETP:n sisään on rakennettu erilaisia työkaluja, jotka ovat joko pakollisia testaukseen liittyviä tai helpottavia ominaisuuksia, jotka edistävät nopeampaa työskentelyä. Kuvassa 1 on ETP:n yleinen toimintaperiaate.



Kuva 1. ETP:n toimintaperiaate.

ETP ei itse kykene testaamaan mitään yksin, vaan tarvitsee erillisiä testilaitteita, joiden kanssa ohjelmisto kommunikoi. Käyttäjää tarvitaan antamaan tarvittavat asetukset ETP:lle, jonka jälkeen tämä osaa kommunikoida haluttujen testilaitteiden kanssa. ETP:n testausjärjestelmä toimii siten, että ensin asetetaan yleiset asetukset kuntoon, joita ovat mm. mittalaitteiden osoitteet ja fyysisten laitteiden asetukset. Tämän jälkeen voidaan lähteä rakentamaan itse sekvenssiä. Sekvenssillä tarkoitetaan tässä tilanteessa, jaksotettua ajoa, jonka sisällä ajetaan halutut testit ja niiden tarkat asetukset.

Sekvenssin luominen onnistuu siihen tarkoitetulla manuaalisella työkalulla, jossa ensin valitaan halutut testattavat laitteet (DUT), niille asetukset ja lopuksi valitsemalla niille ajettavat erilliset testit. Sekvenssien luontiin on myös ETP:n sisäänrakennettu automaattinen sekvenssin luoja. Tällä tarkoitetaan wizard-pohjaista työkalua, joka pyytää tarvittavat tiedot askel askeleelta, ja luo lopuksi käyttäjälle valmiin sekvenssin. Kun sekvenssi on luotu onnistuneesti, se voidaan tallentaa kovalevylle myöhempää käyttöä varten.

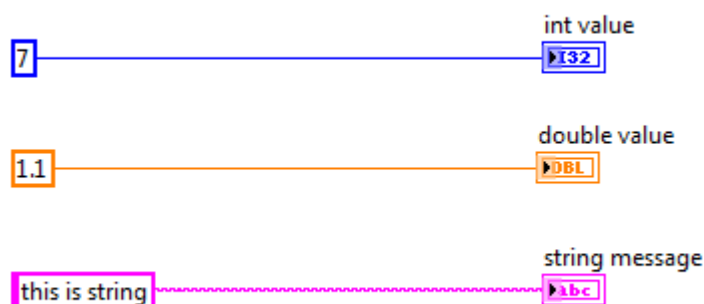
Sekvenssin ajo tapahtuu sillä tarkoitetulla työkalulla. Ensin annetaan tarpeelliset tiedot, jonka jälkeen voidaan suorittaa kyseinen testi. Tietojen syöttökenttiin on rakennettu työkalu, jossa voidaan antaa tarpeellisille tiedoille kiinteät arvot, tällaiset arvot on valmiiksi täytetty, eikä käyttäjän näin ollen tarvitse niitä itse täyttää. Tämän jälkeen käyttäjä on valmis käynnistämään sekvenssiajon, jolloin ETP aloittaa sekvenssin testauksen ja tallentaa samanaikaisesti tuloksia erillisiin testitiedostoihin jatkokäsittelyitä varten.

3 LABVIEW-OHJELMISTO

3.1 LabView yleisesti

LabView on National Instrumentin tekemä kehitysympäristö, joita insinöörit ja tiedemiehet käyttävät graafiseen ohjelmointiin monipuolisten laitteistotukien ansiosta. Tällaisia ovat esimerkiksi erilaiset testauspohjaiset laitteet, joita kytetään käyttämään yhdessä LabView'n kanssa. Tämä takaa nopean suunnittelun ja käyttömahdollisuudet mittaus- ja valvontajärjestelmissä. LabView soveltuu sekä pienille että suurille järjestelmille ja antaa maksimaalisen suorituskyvyn [1].

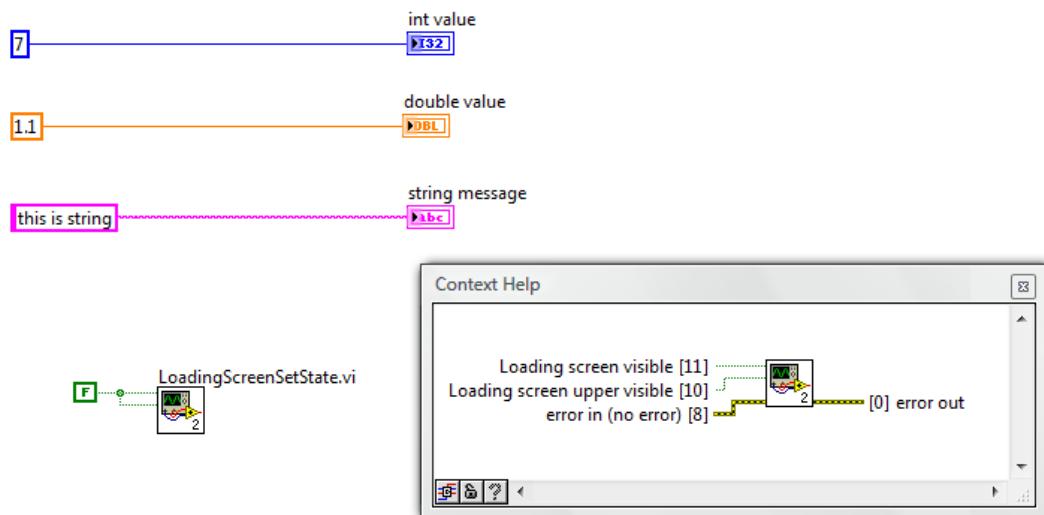
Graafinen ohjelmointikieli on tekstipohjaisesta ohjelmoinnista riippuen pitkälti visuaalista tai sommittelupohjaista ohjelmointia. Graafisessa ohjelmoinnissa tulee vähäلتi käytettyä itse tekstiä ja ohjelmointi suoritetaan erilaisilla valmispalikoilla, jotka ovat suunniteltuja suorittamaan tietyn asian. Yleisesti erilaiset alifunktiot ja operandit yhdistetään ohjelmoidessa keskenään erilaisilla johdoilla, jotka kuvastavat eri tietotyyppjä [1]. Kuvassa 2 on erilaisia tietotyyppi johtoja esiteltynä graafisessa kaaviossa.



Kuva 2. Erilaisia tietotyyppjä graafisessa kaaviossa.

3.2 Virtuaaliohjelmat

Virtuaaliohjelmat (Virtual Instruments) ovat LabView-ohjelmiston ohjelmamoduuleita. Virtuaaliohjelmat sisältävät kaksi eri osaa: etupaneelin sekä graafisen kaavion. Etupaneeli sisältää kaikki käyttäjän tuomat kontrollit, esim. näppäimet ja tekstikentät. Tätä näkymää kutsutaan käyttöliittymäksi ja on myös ohjelmaa suorittaessa näkyvä osa. Graafinen kaavio sisältää taas itse ohjelman eli sen loogisen toiminnan. Haluttaessa virtuaaliohjelmat voivat myös olla aliohjelma, sillä niille voidaan määritellä tulot ja lähdöt [2]. Kuvassa 3 on esimerkki graafisesta kaaviosta, jossa käytetty aliohjelmaa.

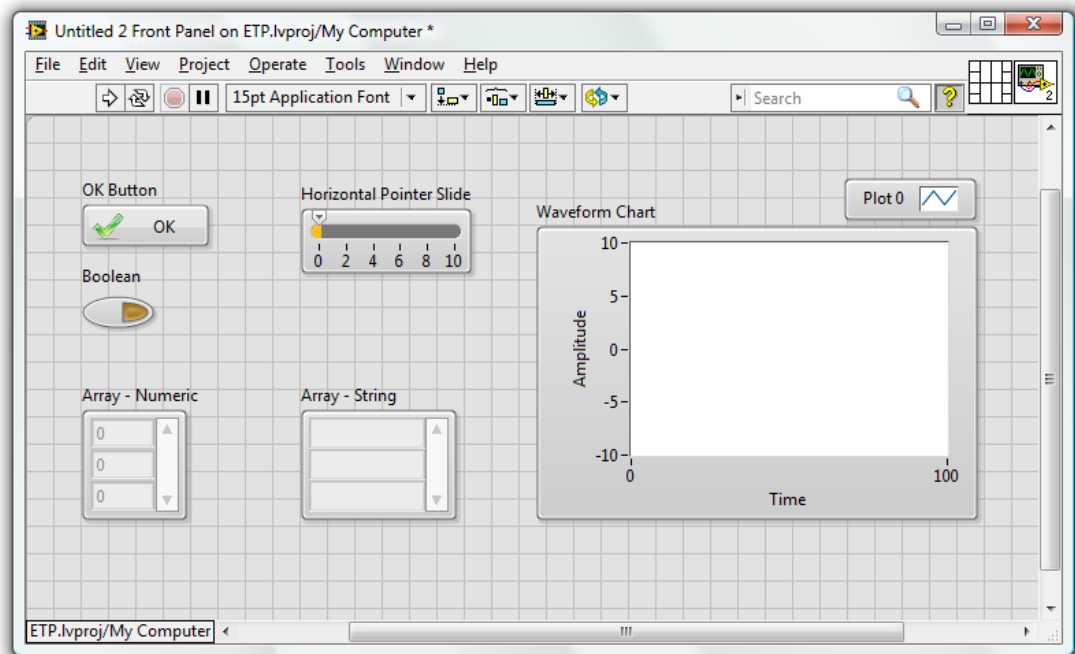


Kuva 3. Graafinen kaavio, jossa käytetty aliohjelmaa.

3.3 Kontrollit

Kontrollit ovat LabView-ohjelmiston omia työkaluja, joita käyttäjä voi lisätä käyttöliittymäänsä, ja muokata näistä oman näköisiä. Kontrollien muokkaaminen onnistuu LabView'n omalla työkalulla sekä itse ohjelmointikoodin kautta. Kontrolleja voisi kuvailla esineiksi, jotka tekevät sitä, mitä niiden on käsketty tekevän. Kontrolleihin kuuluvat mm. näppäimet, tekstinsyöttökentät,

puunäkymä, latauspalkki ja listat ym [2]. Kuvassa 4 on virtuaali ohjelman etupaneelin erilaisia kontrolleja.



Kuva 4. Etupaneeli, jossa esiteltynä LabVIEW'n erilaisia kontrolleja.

4 KÄYTTÖLIITTYMÄN SUUNNITTELU

4.1 Toimiva käyttöliittymä

Toimivan käyttöliittymän rakentaminen ei ole nykymaailmassa helppoa. Käyttöliittymän suunnittelussa täytyy ottaa huomioon ihmisen ja koneen välinen kommunikointi, jotta molemmat ymmärtävät toisiaan mahdollisimman hyvin. Ymmärrys johtaa taas nopeaan ja helppoon käytettävyyteen, jonka johdosta käyttäjä kykenee vapauttamaan kaiken huomion siihen, mitä on tekemässä. Jokaisessa käyttöliittymässä on elementtejä, joita on helppo omaksua, mutta suunnittelun tavoitteena on saada koko käyttöliittymästä sellainen. Jotta käyttöliittymä saataisiin mahdollisimman helposti opittavaksi, on sen oltava myös mahdollisimman yksinkertainen: kaikki tärkeimmät ominaisuudet ovat helposti saatavilla ja niitä voidaan käyttää ainoastaan yhdestä paikasta. Vaikka suunnittelulla on voimakas painostus uutta käyttöliittymää aloitettaessa, on silti muistettava, että aina ei voi ennustaa, miten käyttöliittymä toimii käytännössä. Vaikeasti opittava käyttöliittymä ei ole koskaan käyttäjän vika. Ihmisiä on vaikea muuttaa ja yleensä ongelmat tulevat aikaisemmista käyttöliittymäkokemuksista [3].

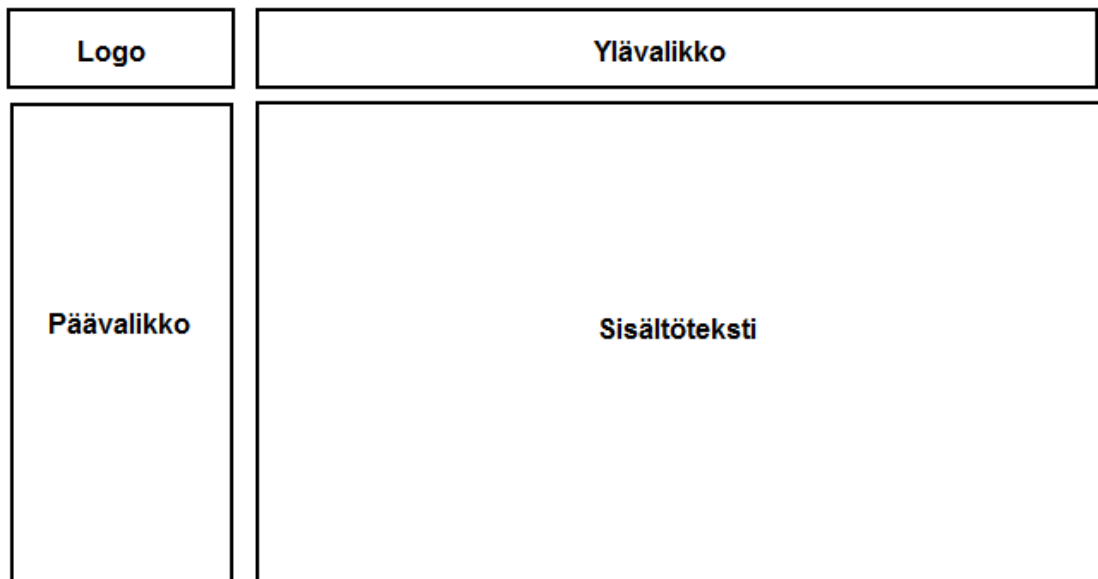
ETP:n käyttöliittymän suunnittelu aloitettiin tutustumalla samantapaisten testilaitteiden tai vastaavien ohjelmistojen toteutuksiin. Tällä keinolla pyrittiin hyödyntämään hyviä ideoita ja jalostamaan niitä käytettävyyden parantamiseksi. Ohjelmiston yleisilme tuotiin esille rautalankamallien avulla, jotka tehtiin tietokoneella. Rautalankamallien jälkeen lähdettiin suunnittelemaan eri ominaisuuksien toiminnallisuutta ja käytettävyyttä.

4.2 Rautalankamallit

Rautalankamallia käytetään nykyään monissa eri projekteissa. Näillä hahmotetaan yleiskuvaa siitä, millaisia asioita halutaan tuoda ohjelmistossa esille avaintekijöinä. Rautalankamalli on joko kytkentäkaaviomainen tai epätarkka kokonaisuus tuotteen käyttöliittymästä, jonka ensisijainen tarkoitus on osoittaa erilaisia toimintoja, ominaisuuksia, sisältöä sekä käyttäjän kokemusta ilman tarkkoja kuvauksia tuotteesta. Rautalankamallinnuksia voidaan käyttää monissa erityyppisissä ratkaisuissa, kuten tietokoneohjelmistoissa, erilaisissa web-sivustoissa tai yleisesti kaikissa tuotteissa, joissa ollaan tekemisissä ihmisen ja tietokoneen välillä [4].

Rautalankamalleja voidaan kutsua myös ns. pohjapiirustuksiksi tietystä tuotteesta, mutta niitä ei ole normaalisti tarkoitettu esittämään tuotteen graafista suunnittelupuolta laisinkaan. Rautalankamallit luodaan yleensä aivan projektin alkuvaiheessa. Rautalankamallit ovat voimakas työkalu esittämään, millaista tuotetta ollaan rakentamassa, josta projektin muutkin osapuolet esim. suunnittelijat, ohjelmoijat sekä tuotepäälliköt, näkevät kokonaisuuden ja voivat antaa ideoita ennen itse työn varsinaista aloittamista [5].

Rautalankamallit voidaan tehdä tietokoneella, mutta myös paperille luotu malli on tehokas. Karkeana esitysmuotona voidaan tehokkaasti esitellä käyttöliittymän osia, jotka voidaan tarpeen tullen nopeasti hävittää ja uudelleen piirtää uusien ideoiden mukaisesti, kunnes lopulliseen ratkaisuun on päädytty. Yleisesti rautalankamalli sisältää pelkästään yksinkertaisia viivoja, jotka vain kuvastavat tietyn osan kohdistamista itse ohjelmistossa. Näin saadaan unohdettua visuaalinen ilme tässä vaiheessa projektia ja tuotua jokaisen osapuolen mieleen oma näkemys siitä, mikä olisi tärkeintä ja mikä taas ei [4]. Kuvassa 5 on tehty perinteinen rautalankamalli web-sivustosta tietokoneen avulla.



Kuva 5. Perinteinen rautalankamalli web-sivustosta.

4.3 Käytettävyys

Onnistuneessa käyttöliittymässä kokonaisuus on elintärkeää. Käyttöliittymän tarkoitus on tuoda käyttäjälleen kokonaisvaltaisen kuvan, jossa näppäimet, navigaattorit, kuvat, logo, tekstit ja otsikko luovat yhtenäisen rakenteen, jota on käyttäjän mukava katsella ja helppo käyttää. Käyttöliittymä on siis käyttäjän ja tuotteen välinen työskentelyalue [6].

Pystyäkseen taas käyttämään tuotetta on käyttäjän ymmärrettävä edellä mainitut asiat sekä ymmärtämään kaikki oleellinen sen toiminnasta. Käytettävyydellä on tässä asemassa iso rooli, mikäli käyttäjä haluaa päästä tavoitteeseen mahdollisimman nopeasti ja vaivattomasti [7].

Käytettävyydellä kuvastetaan tuotteen toimivuutta. Jos tuotetta on helppoa ja tehokasta käyttää ja se suorittaa sen tehtävän mitä sen pitää, on tällä hyvä käytettävyys.

Käytettävyydellä on myös monia tärkeitä standardeja, joita tulisi noudattaa, jotta käytettävyydestä saisi mahdollisimman toimivan ja tehokkaan:

Näkyvyys järjestelmän tilasta

Järjestelmän tulisi aina pitää käyttäjä ajantasalla, siitä mitä ruudulla tapahtuu kohtuullisessa ajassa. Tällä keinolla päästään eroon ongelmasta, jossa käyttäjä luulee ohjelman suorittamisen loppuneen tai ongelmatilanteen, jossa ohjelma olisi virheellisesti sammunut [8].

Kommunikointi käyttäjän ja koneen välillä

Järjestelmän tulisi ilmoittaa käyttäjälle sellaista kieltä, sanoja, lauseita tai käsitteitä, joita ihminen voisi ymmärtää. Kommunikointi tulisi kulkea loogisessa ja luonnollisessa järjestyksessä koneen ja käyttäjän välillä [8].

Käyttäjän vapaa kontrollointi järjestelmään

Käyttäjät valitsevat usein erilaisia järjestelmän toimintoja vahingossa, jonka ansiosta tarvitsevat "häätäuloskäynnin". Tämän takia toimintojen "kumoa" tai "tee uudelleen" toiminnot olisi suotavaa olla käyttäjän valittavissa [8].

Johdonmukaisuus ja standardit

Käyttäjän ei tulisi koskaan ihmetellä erilaisia sanoja, toimintoja tai tilanteita, jotka tarkoittavat samaa asiaa. Näin ollen tulisi käyttää johdonmukaisuutta yleissopimuksilla ja käyttää näitä joka puolella järjestelmää [8].

Virheiden ehkäisy

Virheilmoitukset ovat tärkeitä järjestelmissä, mutta tehokkaampaa olisi ennaltaehkäistä kyseiset tilanteet kokonaan. Käyttäjälle tulisi antaa selkeät ohjeet tarpeellisista ehdoista ja esittää ne käyttäjälle vahvistuksena ennen sitoavaa toimenpidettä [8].

Tunnistaminen muistamisen sijaan

Käyttäjän muistia tulisi kuormittaa mahdollisimman vähän sijoittamalla esineet, toimet sekä vaihtoehdot mahdollisimman helppoihin paikkoihin. Käyttäjän ei tulisi jatkuvasti etsiä tietoa, mistä jokin tietty toimenpide tehtiin vaan löytää se ilman suurempia ongelmia. Näin ollen opasteiden tulisi olla näkyvissä tai helposti saatavilla aina tarvittaessa [8].

Joustava ja tehokas käyttö

Kokenut käyttäjä pystyy nopeuttamaan käyttöä osaamisensa ansiosta, mutta uuden käyttäjän kannalta järjestelmän tulisi olla joustavasti rakennettu, jotta tehokas käyttö olisi mahdollisimman helposti saavutettavissa. Usein järjestelmän räätälöinti edesauttaa joustavaan ja tehokkaaseen käyttöön, juurikin sen takia, että käyttäjä saa siitä oman näköisensä [8].

Esteettinen ja selkeä muotoilu

Graafisen käyttöliittymän tulisi olla mahdollisimman selkeä ilman ylimääräisiä ja merkityksettömiä asioita. Näin saadaan selkeytettyä käyttäjälle vain tarpeellisten toimintojen, kuten nappuloiden näkyvyyttä [8].

Ohjastaa käyttäjiä tunnistamaan, diagnosoimaan sekä toipumaan virheistä:

Virheilmoitukset tulisi ilmaista selkokielellä, jotka nimenomaan osoittavat ongelman ja opastavat virheilmoitukseen ratkaisua [8].

Ohjeet ja dokumentaatio

Järjestelmää tulisi voida käyttää ilman asiakirjoja, mutta silti opasasiakirjojen tulisi olla tehtynä. Näin saadaan helposti koottua laajempi kokonaisuus järjestelmästä, joita ei onnistuta esittämään tehokkaasta itse ohjelmistossa. Tällaisten asiakirjojen tulisi olla helposti saatavilla, jotta käyttäjä löytää etsimänsä mahdollisimman helposti [8].

4.4 Graafinen suunnittelu

Graafinen käyttöliittymä perustuu käyttöliittymäelementteihin, joilla käyttäjä käyttää ohjelmistoa. Käyttöliittymäelementit koostuvat erilaisista ikkunoista, kuvakkeista, näppäimistä ja valikoista [9]. Graafisen käyttöliittymän suunnittelussa on suositeltavaa käyttää yhdenmukaisuutta. Näin saadaan selkeä ja helppolukuinen lopputulos. Yksinkertaisuus on avaintekijä graafisen käyttöliittymän toteutuksessa, joka takaa tehokkaan oppimisen sekä antaa käyttäjälle enemmän vaihtoehtoja edetä tavoitteeseensa nopeammin. Graafisen käyttöliittymän toteutus on täysin erilainen mobiililaitteilla kuin työpöytäsovelluksilla. Yleisesti mobiililaitteilla tulevat ensimmäisenä vastaan kokorajoitukset, jotka tuottavat täysin eri lähtökohdat käyttöliittymän luomista varten. Mikäli graafinen suunnittelu on tehty huolella, on käyttäjä helposti ohjattavissa tämän haluamiin paikkoihin. Huonolla graafisella suunnittelulla voidaan pahimmissa tapauksissa ajaa käyttäjä käyttämään vaihtoehtoisia ohjelmia [10].

4.5 Värimaailma

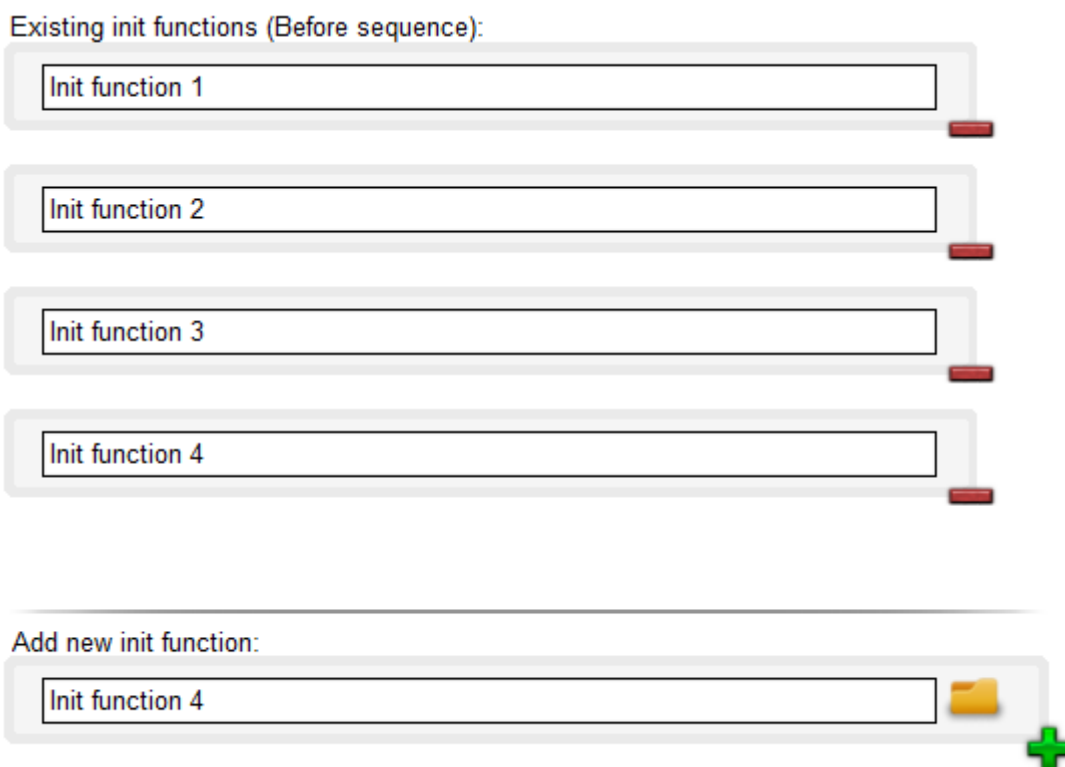
Käyttöliittymän värimaailma on tärkeä, jotta käyttäjä pystyy havaitsemaan kaikki tarpeelliset ominaisuudet käyttöliittymästä. Käytön pitää olla näin ollen helppoa sekä luontevaa ihmiselle, jotta haluttuun lopputulokseen päästään. Värien valinta on paljolti mielipidekysymys, mutta monia, tehokkaiksi havaittuja värimallipohjia on olemassa, ja ne kannattaa ottaa huomioon käyttöliittymää luodessa. Värejä ei myöskään tulisi käyttää liian monia, jotta lopputuloksesta ei tulisi epäselvä, joka vaikeuttaa käyttäjää hahmoittamasta kokonaisuutta. Hyvä käyttösuhde erilaisten värien määrässä vaihtelee 2-5. Taustaväreinä ei tulisi käyttää vahvoja tai tummia värejä, jotka vievät huomion muista elementeistä käyttöliittymässä. Vaaleat taustavärit taas edesauttavat muiden elementtien korostumista pohjaväristä [11].

5 KÄYTTÖLIITTYMÄN TOTEUTUS

5.1 ETP:n ongelmat ja haasteet

ETP:n uuden käyttöliittymän suunnittelu ja toteutus aloitettiin yleisen ulkoasun uudelleen suunnittelulla, uusien ominaisuuksien ideoinnilla ja vanhojen ominaisuuksien paranteluilla. Ohjelmassa ei siis ollut yleistä ulkoasua vaan kaikki oli sijoiteltu ajanmittaa omiin ikkunoihin ja osa ominaisuuksista oli oudosti ja hajanaisesti sijoiteltu. Myös graafisella puolella oli käytetty monia erilaisia tyylejä, mikä hajoitti kokonaisuutta.

Näiden haasteiden korjaus toi kaksi ongelmaa: ETP:n käyttöliittymä oli liian monimutkainen, joten siitä piti saada yhtenäinen kokonaisuus. Kaikki ominaisuudet ja asetukset toimivat omissa ikkunoissa, joka hajotti käyttäjän ymmärrystä kokonaisuudesta. Monia ominaisuuksia oli jätetty täysin selittämättä ja niiden käytöstä oli monia käsityksiä. Ensimmäistä kertaa liittymää käyttäessäni ymmärsin väärin monet työkalut niiden monimutkaisuuden vuoksi. Myös paikasta A paikkaan B klikkauksien määriä ryhdyttiin tarkkailemaan, tarkoituksena saavuttaa lisänopeutta käyttäjälle, jonka tavoitteena oli saada haluttu toiminto tehtyä sulavasti ilman turhia kiertoteitä. Näistä klikkauksista saatiin parhaimmillaan pois jopa 80 % vanhaan nähden. Myös listapohjaisiin työkaluihin saatiin yhtenäinen toimivuus, joka antaa käyttäjälle varman olon siitä, mitä oikein ollaan tekemässä. Kuvassa 6 on listapohjainen ominaisuus esiteltynä: käyttäjä kykenee lisäämään halutun polun vihreällä plus-napilla ja tämän jälkeen poistamaan niitä halutessaan ylemmästä listasta punaisella minus-napilla.



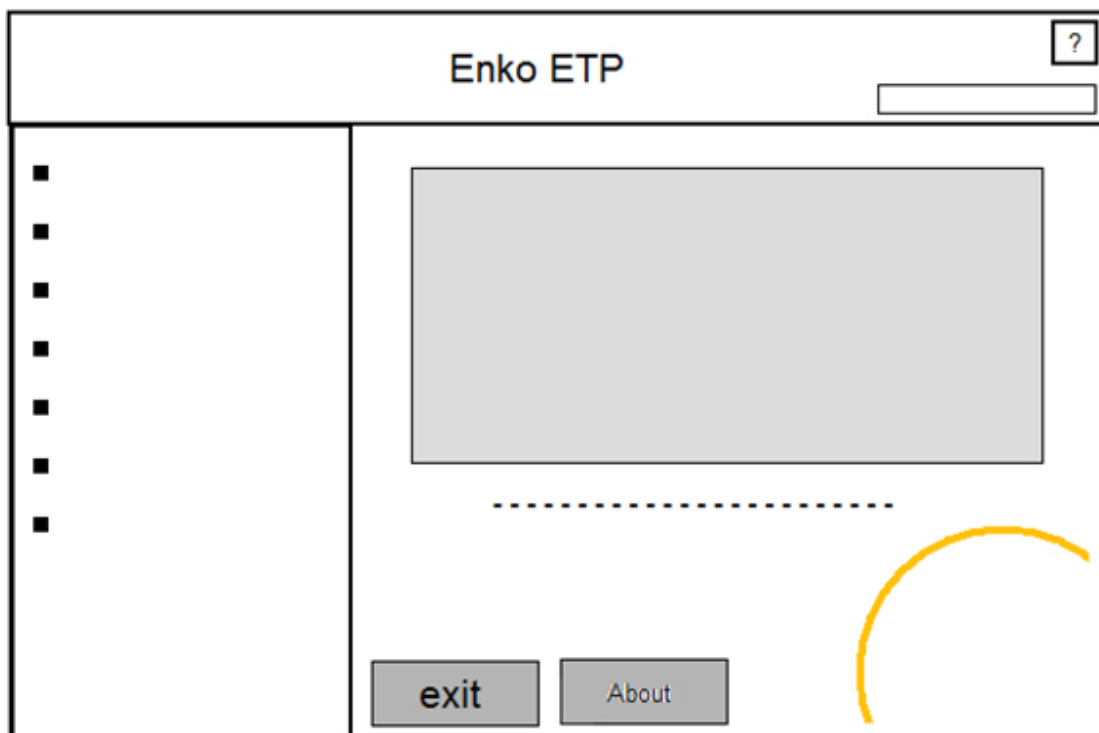
Kuva 6. Listapohjaisen ominaisuuden vihreät ja punaiset napit.

Toisena ongelmana oli, että graafinen ulkoasu oli aikakautensa lopussa. Vanha ulkoasu ei enää täyttänyt nykyajan standardeja, joten graafinen toteutus päätettiin aloittaa täysin puhtaalta pöydältä. Ulkoasun muotoilussa ainoana kriteerinä oli käyttää TestHouse Enko Oy -yrityksen omia värejä, jotka olivat musta, oranssi ja näiden eri värisävyt. Näin ollen sain siis vapaat kädet luoda omantyylistä grafiikkaa yrityksen standardeja unohtamatta. Muotoilun vaiheita käytiin viikoittain yhdessä läpi ja lopullinen ulkoasu hyväksyttiin Enkon toimesta.

5.2 Yleistä toteutuksesta

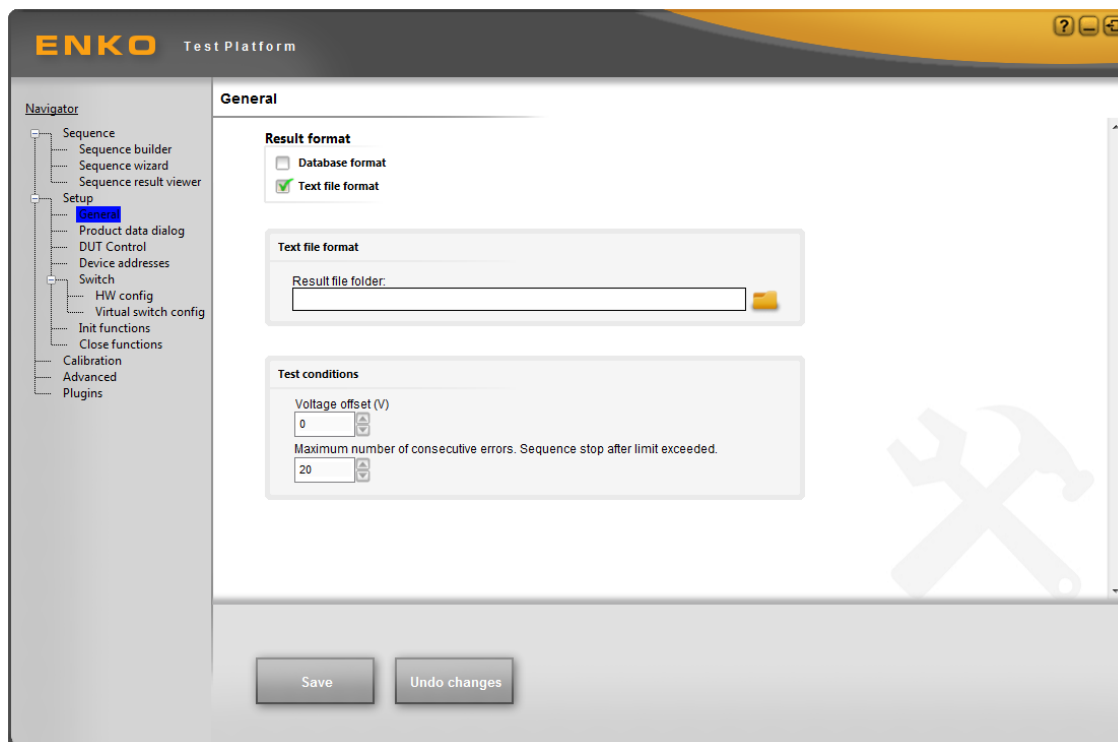
ETP:n käyttöliittymän toteutus aloitettiin ryhmäkeskustelulla, millaista yleisilmettä haettiin, mitä värejä sai käyttää ja mitä materiaaleja oli jo yrityksen omasta takaa. Alun jälkeen tutustuttiin vanhaan ohjelmistoon ja sen käyttöliittymään, jolloin saatiin käsitys siitä, miten ohjelmisto toimi. Tämän jälkeen tutustuin LabView'hin, jolla koko ohjelmisto tehtiin.

Toteutus aloitettiin rautalankamalleista, joilla sai hahmotettua erilaisia variaatioita omista näkemyksistään. Nämä mallit myös autoivat hyvin aukaisemaan kokonaisuutta myös toisille projektissa olleille henkilöille. Kuvassa 7 on ensimmäinen otos ETP:n etusivun rautalankamallista, jonka toteutus on hieman erilainen viimeiseen versioon nähden.



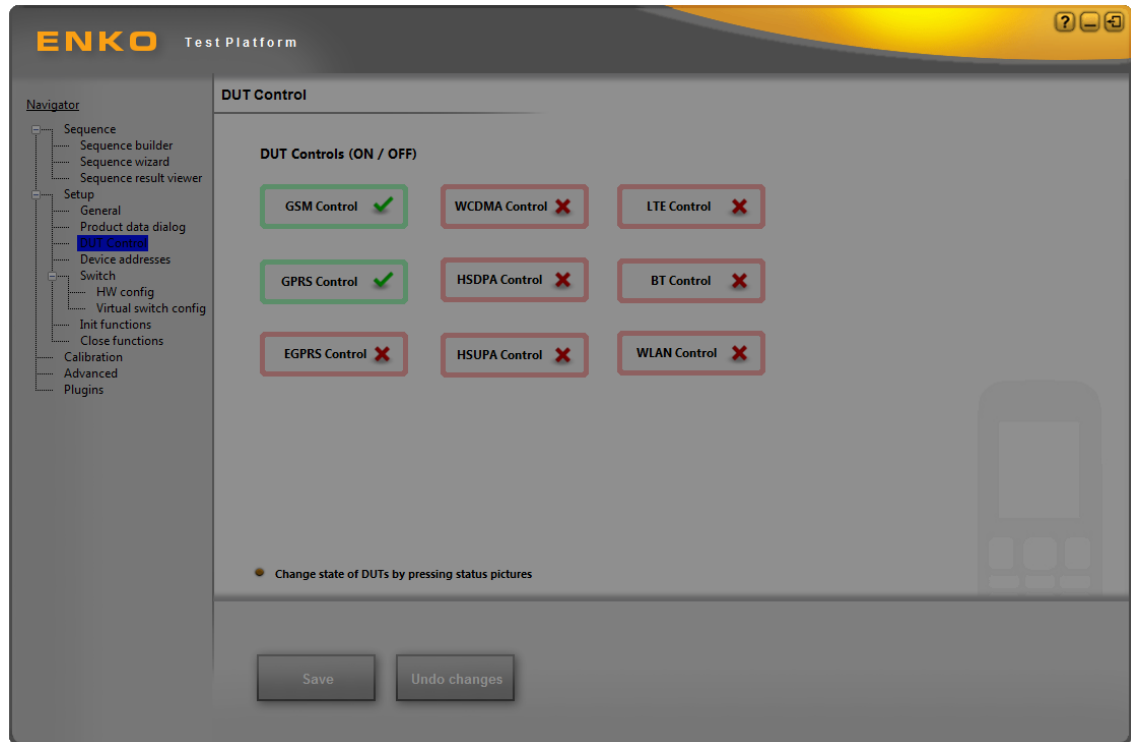
Kuva 7. Ensimmäinen rautalankamalli ETP:stä.

Graafinen toteutus luotiin GIMP-ohjelmistolla, joka on täysin ilmainen. Suurimpana ongelmana kyseisen ohjelmiston kanssa olivat erilaiset työkalut, pikanäppäinyhdistelmät sekä yleisien ominaisuuksien uupuminen verrattuna vastaaviin kilpaileviin kuvankäsittelyohjelmistoihin. Ensimmäiset vedokset rakennettiin rautalankamallien pohjalta, joista edettiin suuremmista osista aina pienempiin palasiin, joista syntyi yhtenäinen ja hallitseva kokonaisuus. Kuvassa 8 on käyttöliittymän ulkoinen olemus.



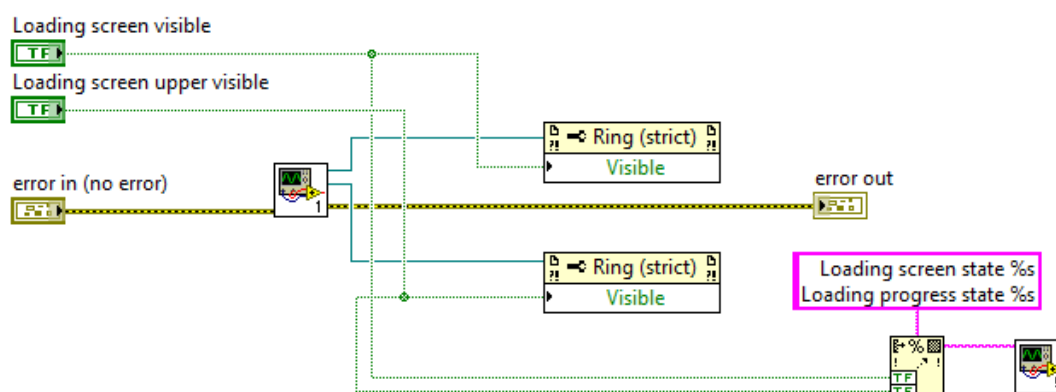
Kuva 8. Käyttöliittymän ulkoinen olemus.

Ohjelmistototeutus tehtiin LabView-ohjelmointiympäristöllä, jonka opettelu oli luultua haasteellisempaa sen monipuolisten ominaisuuksien ansiosta. Osa työkaluista oli myös monimutkaisia, jolloin näiden työkalujen hyödyksi käyttäminen oli oletettua haastavampaa. Ohjelmointipuoli oli pitkälti jo tehty, mutta muutamia uuden käyttöliittymän tuomia ominaisuuksia jouduttiin ohjelmoimaan alusta alkaen. Vanhassa käyttöliittymässä ei ollut laisinkaan latausikkunaa, joten sellainen oli ehdotonta rakentaa, kuten kuvasta 9 voi sen havaita. Latausikkunaa indikoi tummempi ruutu, joka estää käyttäjän tekemästä ei-sallittuja toimenpiteitä ja ylhäällä oleva oranssi latausanimaatio, joka pyörii niin kauan, kunnes lataus on suoritettu.



Kuva 9. ETP:n latausikkuna.

ETP:n latausikkunan tekninen toteutus aloitettiin luomalla harmaa pohja sekä oranssi animaatioindikoija, joilla havainnollistettiin, että lataus oli käynnissä. Näistä kahdesta osasta tehtiin erilliset kontrollit, joita kyettiin komentamaan itse koodin puolelta. Latausikkunasta tehtiin erillinen virtuaaliohjelma, jota voitiin halutessa kutsua milloin tahansa koodia suorittaessa, kuten kuvasta 10 voidaan todeta. Näin saatiin nopea ja helppo tapa näyttää latausruutu tarvittaessa. Kuvassa 10 on esiteltyä latausikkunan graafinen kaavio sen omasta virtuaaliohjelmasta.



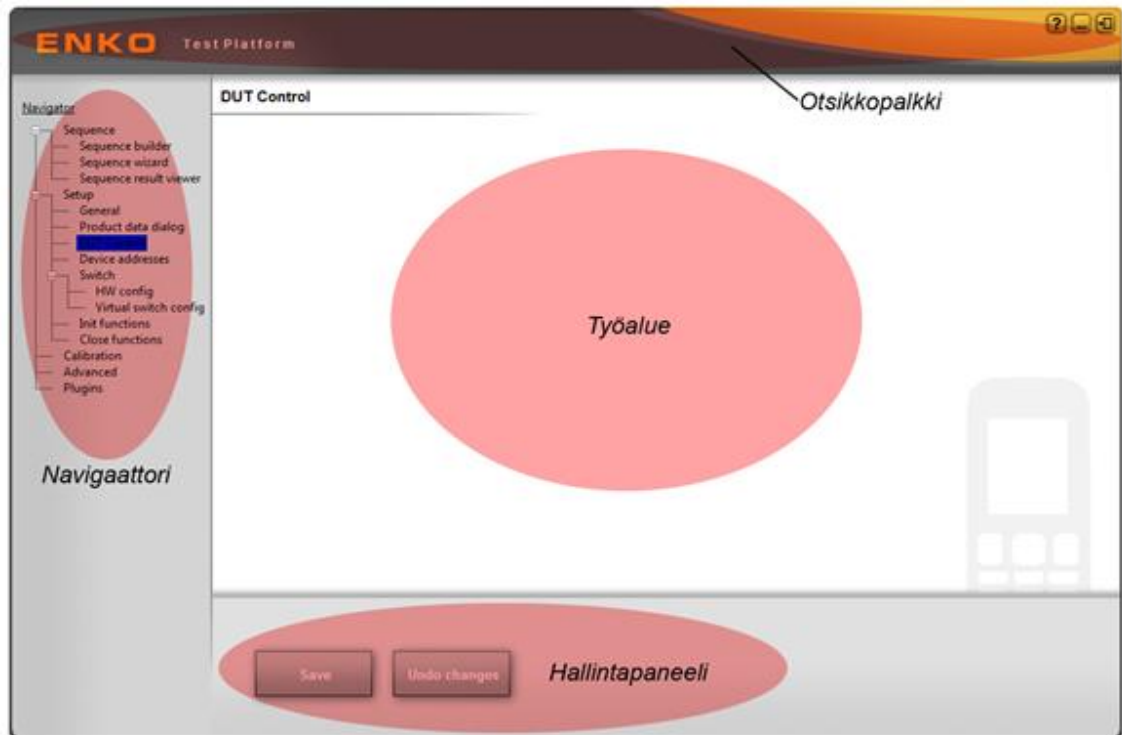
Kuva 10. Latausikkunan graafinen kaavio.

5.3 Ominaisuudet

Osa ETP:n ominaisuuksista rakennettiin täysin uusiksi vanhojen nojalta. Tällaisiin ratkaisuihin päädyttiin, kun ominaisuus ei kertonut käyttäjälle, mitä sillä voitiin helposti tehdä. Tällaisia oli mm. listojen käsittelyt, joihin kyettiin lisäämään ja poistamaan annettuja arvoja, sekä siirtämään niitä haluttuun juoksevaan järjestykseen, mikäli tarpeen.

5.4 Graafinen toteutus

ETP:n kannalta graafiseen käyttöliittymään valikoitiin yksinkertainen, mutta tehokas kokonaisuus. Kaikki tärkeät asiat, kuten näppäimet, ikkunat ja tekstikehykset ovat sopivan isoja ja selkeitä. Näin ollen käyttäjä ei hämmenny, mikä kyseinen kontrolli on ja mikä sen tehtävä on ohjelmassa. Ohjelmistossa on myös käytetty samankaltaista tyyliä kaikkialla, jottei käyttäjä luule päätyneensä täysin eri palveluun. ETP:n graafinen liittymä tukee myös sen värimaailmaa tarjoamalla sopivan muotoisia ja tyyliä ratkaisuja kokonaisuuteen nähden. Kuvassa 11 on graafisen käyttöliittymän kokonaisuus esiteltyä.



Kuva 11. Käyttöliittymän kokonaisuus esiteltynä erillisinä osioina.

5.5 Käyttöliittymän rakenne

ETP:n vanha käyttöliittymän rakenne oli ns. tilkkutäkki. Monia uudistuksia oli tullut vuosien saatossa, mutta kaikki yksityiskohdat oli huonosti sijoitettu omiin paikkoihinsa. Suurin ongelma vanhassa käyttöliittymässä oli "uudet ikkunat". Aina kun mikä tahansa valikko avattiin, aukesi se omaan ikkunaansa jättäen käyttäjän miettimään mitä seuraavaksi. Monia asia tarvitsi huoltoa ja näin ollen tehtiin yhdessä ikkunassa toimiva kokonaisuus, jota kyettiin kontrolloimaan navigaattorilla.

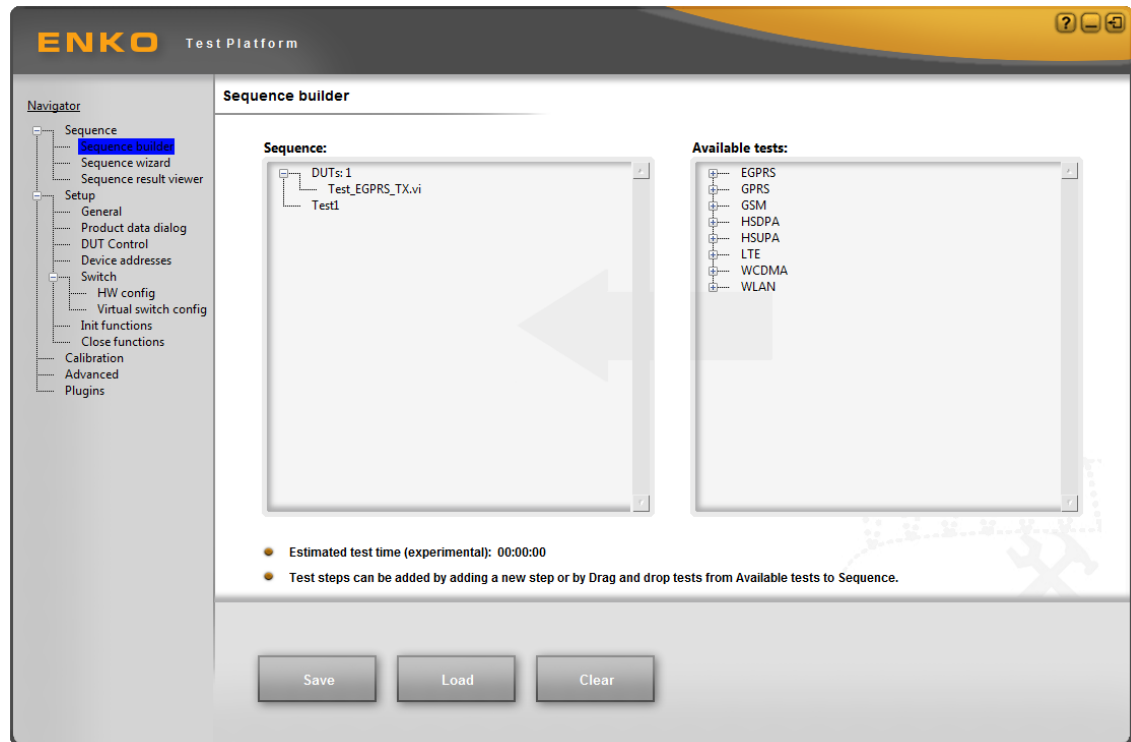
ETP:n uuteen käyttöliittymään panostettiin, jotta kaikki oleellinen ja tärkeä saatiin mahdollisimman näkyviin ja nopeasti ymmärrettäväksi. Ylimääräiset ja turhat ominaisuudet karsittiin pois ja tilalle tuotiin yksinkertaisempia vaihtoehtoja niin graafisesti kuin toiminnallisesti. Yleisesti käyttöliittymän rakenne muodostui otsikkopalkista, työalueesta, navigaattorista ja hallintapaneelistä.

Otsikkopalkki, joka sijaitsee yläpäänä, on mahdollisimman selkeä, jotta saadaan käsitys siitä, minkä kokoinen ohjelma on. Tällä viestitetään käyttäjälle muiden ohjelmien tapaa, mistä sitä voi liikuttaa. Tuttuun tapaan myös ohje sekä pienennä- ja sammuta-napit löytyvät oikeasta yläkulmasta. Nämä ovat otsikkopalkin nappeja, joilla voi hallita itse ohjelmaa milloin ja missä tahansa. Otsikkopalkista voidaan myös siirtää ETP:tä työpöydällä käyttäjän mielen mukaisesti. Kuvassa 12 on otsikkopalkki esiteltynä.



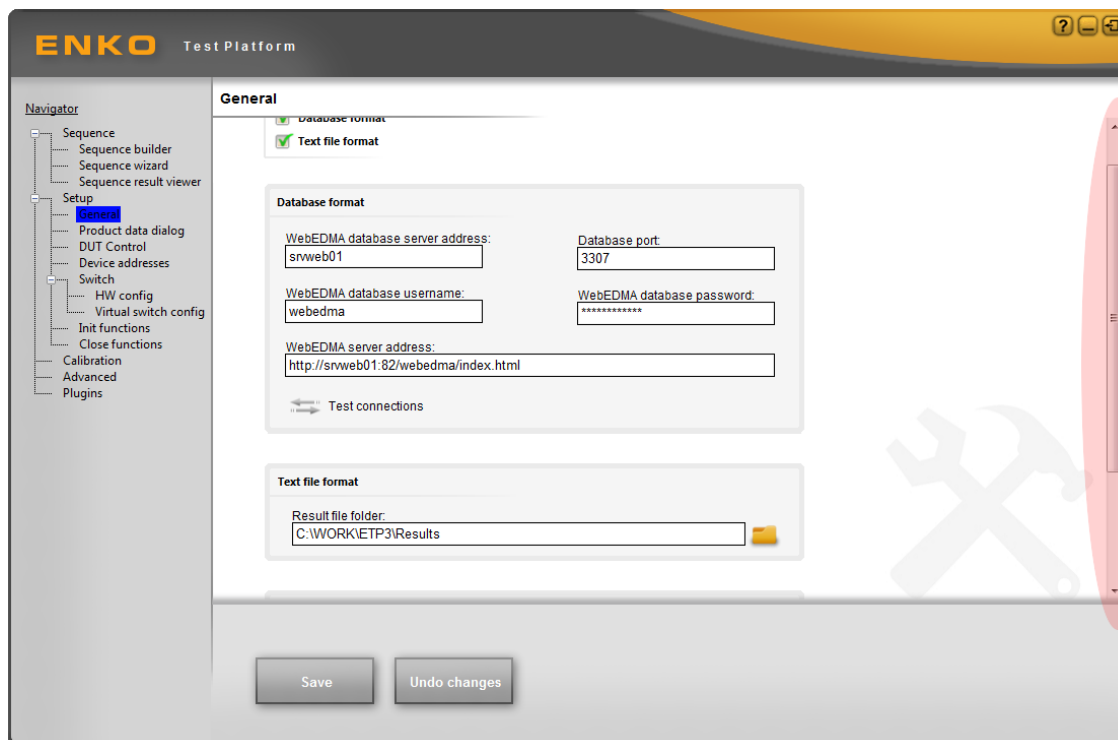
Kuva 12. Otsikkopalkin esittely.

Työalue sijoitettiin mahdollisimman keskelle ja isona. Näin ollen käyttäjälle ei jää epävarmuuksia, missä itse työ tehdään ja minne kaikki uudet asiat tulostuvat. Työalueella ilmaistaan aina myös, missä käyttäjä parhaillaan sijaitsee, jotta epäselvyyksiä ei pääse tapahtumaan. Työalueella käyttäjä pystyy tekemään tarpeelliset toimenpiteet, jotta mahdollinen työn tavoite saadaan valmiiksi. Kuvassa 13 ETP:n työalue esiteltynä sekvenssin rakentaja työkalusta.



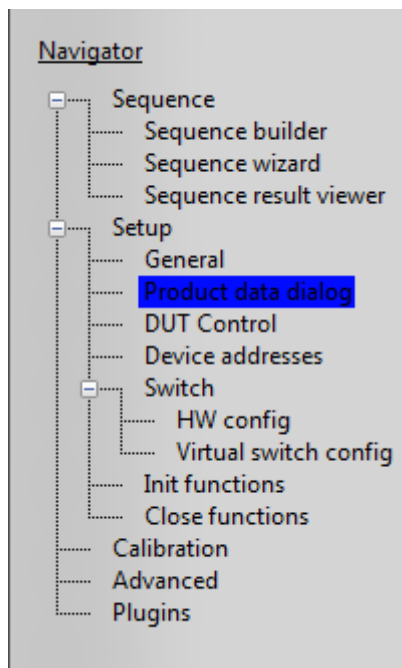
Kuva 13. ETP:n työalue esiteltyinä sekvenssirakentajan avustuksella.

Työalue on suunniteltu siten, että kaikki oleellinen mahtuu ruudulle. Mikäli tarpeellista tietoa tulee enemmän kuin ruudulle mahtuu, ilmestyy vierityspalkki oikealle sivuille, mikä mahdollistaa työalueen vierittämistä ales nähdäkseen kaikean mahdollisen tiedon. Kuvassa 14 esiteltyinä ilmestyvät vierityspalkki, mikäli tietoa tulee enemmän kuin työalueen ruudulle pystytään tulostamaan kerralla.



Kuva 14. Vierityspalkin tulostus, mikäli tarpeellinen.

Navigaattori perustuu ns. puunäkymään. Puunäkymän perusidea on nostaa tärkeimmät asiat ylimmiksi oksiksi ja vähemmän tärkeämmät alemmille oksille. Tämä sama periaate toimii täydellisesti ETP:n kanssa. Tärkeimmät työkalut, joita käytetään eniten, ovat korkeimmilla oksilla ja näiden asetukset ovat taas alemmilla. Näin käyttäjä oppii nopeasti etsimään tarpeellisen työkalun mahdollisimman vähällä vaivalla. Toinen navigaattori sijaitsee ohjelmassa täysin alalaidassa, jossa sijaitsee kaikki sillä hetkellä käytettävät päänappulat. Tämä navigaattori vaihtaa nappuloitaan riippuen siitä, mitä työkalua parhaillaan käytetään. Päänappuloilla tarkoitetaan kontroleja, joilla voidaan hallita sen hetkistä työaluetta. Kuvassa 15 on ETP:n navigaattori puunäkymänä esiteltyinä.

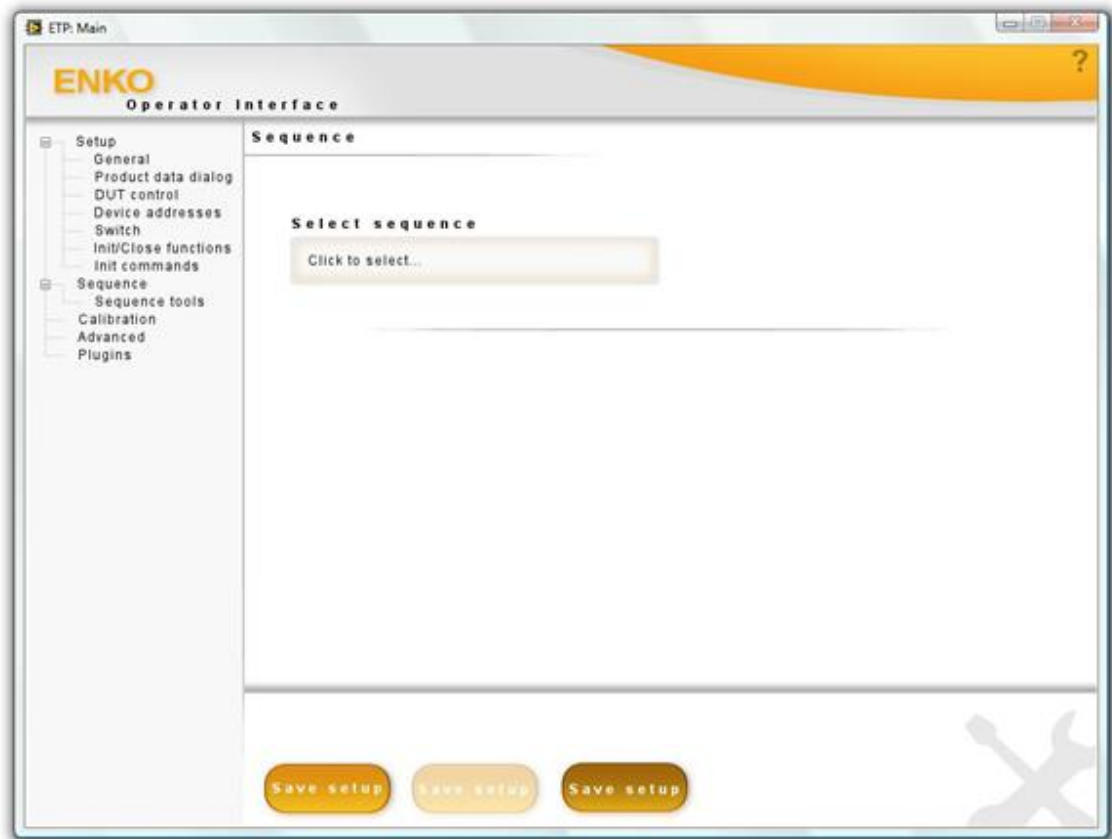


Kuva 15. ETP:n puunäkymä.

Käyttöliittymän käytettävyyttä parannettiin yhdistämällä kaikki ylimääräiset ikkunat yhdeksi ja yhdistämällä vanhat ominaisuudet samoihin paikkoihin, mikäli näin parhaaksi todettiin. Käytettävyyden kannalta pidettiin joitakin hyväksi todettuja (drag & drop) ominaisuuksia tietyissä työkaluissa, mutta niihin annettiin enemmän selkeyttä. Näin ollen kunnioitettiin vanhoja tuotteen käyttäjiä, sekä auttamaan uusia käyttäjiä ymmärtämään tuotetta paremmin ja käyttämään sitä tehokkaammin.

5.6 Värimaailman toteutus

ETP:n värimaailmasta sovittiin työnantajan kanssa jo heti suunnitteluvaiheen alussa. Suositeltavaa oli käyttää yrityksen omia päävärejä (oranssi/musta) sekä muita tummia sävyjä vaaleiden sijaan. Ensimmäiset suunnitteluvaihdokset tehtiin kuitenkin vaaleammilla sävyillä kokeilumielessä, mutta näiden vaihtojen jälkeen päätettiin tummentaa sävyjä kokonaisvaltaisesti. Näin graafinen toteutus tuki myös Enkon värimaailmaa, joten se ei eroa yrityksen yleiskuvasta, mikä taas tuo enemmän yhteneväisyyttä yrityksen keskuudessa. Kuvassa 16 otos vaaleasta pohjasta, jonka toteutusta ei tultu jatkamaan.



Kuva 16. ETP:n vaaleapohja.

Yleisesti viimeisimmän ETP:n värit muodostavat sopivalla kontrastilla erottuvat pääasiat. Otsikkopalkki-alue, jonka värit ovat tummimpia koko ohjelmassa, navigaattorit, jotka ovat vaaleanharmaan sävyisiä, ja työalue, joka on vaalein koko ohjelmistossa. Tämä kokonaisuus todettiin parhaaksi käyttäjän kannalta juurikin sen selkeyden ja tuttujen normien ansiosta.

6 KÄYTTÖLIITTYMÄN TESTAUS

Käyttöliittymän testauksella tuodaan esille epäselvyydet, ristiriitaisuudet ja huonot vuorovaikutusmekanismit, joita testaajat löytävät käyttämällä itse ohjelmistoa. Käyttöliittymä voidaan jakaa useampaan testiosuuteen, jotta mahdolliset ongelmat löytyvät tehokkaasti sekä nopeasti. Käyttöliittymän yleiset ominaisuudet voidaan testata niin, että kaikki mahdollinen tulostuu ruudulle oikein ja halutulla tavalla, jotta näkyvä sisältö on käyttäjän tavoitettavissa ilman näkyviä virheitä. Tällaisia ovat esimerkiksi navigointi, jolla voidaan liikkua ohjelmistossa, paikasta a paikkaan b. Yleiset ominaisuudet pitävät sisällään ohjelman kehukset, otsikkopalkin, navigaattorin hallintapaneelin, työalueen sekä muut ohjelman työkalut, joiden pitää olla toiminnassa aina kun ohjelmaa suoritetaan tietyssä vaiheessa. Yleisessä testauksessa on hyvä ottaa huomioon myös vaihtuvat resoluutiot. Tämän avulla poistetaan virhetilanteet, joita voi ilmetä, mikäli käyttäjällä on liian pieni resoluutio käytössä, joka taas estää ohjelmiston toimivasta sille kuuluvalla tavalla [12].

ETP:n päätestauksen suoritti 3 henkilöä, joilla oli aikaisempaa kokemusta vanhasta ETP:stä. Tällä tavoin saatiin tehokas testausympäristö jo valmiiksi oleville ominaisuuksille sekä työkaluille, joita ETP sisälsi jo ennen käyttöliittymän päivittämistä. Käyttöliittymän testaus toteutettiin samalla kun ohjelmistoa rakennettiin, eli kaikkia ominaisuuksia ei välttämättä ollut vielä lisätty ETP:hen, jolloin osa testattavista ominaisuuksien osista käytiin läpi vasta niiden valmistuttua. Osa ominaisuuksista taas testattiin heti, kun ominaisuus oli saatu valmiiksi ja saatu lisätty ETP:hen. ETP:n testausta suoritti myös 2 muuta henkilöä, jotka eivät olleet aikaisemmin käyttäneet ohjelmistoa laisinkaan. Näin saatiin kokonaisvaltainen testaus, jossa olivat mukana niin itse ETP:n hyvin tunteneet henkilöt ja myös sitä vähemmän käyttäneet henkilöt. ETP:n kannalta testausta aina riittää, koska uusia ominaisuuksia tullaan lisäämään asiakkaiden tarpeiden mukaisesti. Tällaisissa tilanteissa testaus tullaan suorittamaan ominaisuuden lisäämisen jälkeen, jolloin käydään kaikki tilanteet läpi, mitä

kyseisellä ominaisuudella voidaan tehdä. Tällä keinolla saadaan suurimmat virhetilanteet korjattua heti kättelyssä.

Ohjelman kehyksillä tarkoitetaan itse ohjelmaa, jonka sisällä kaikki tapahtuu. Testauksella saadaan tuotua esille ongelmia, jotka saattavat estää ohjelmiston käynnistymisen tai käynnistymisen jälkeiset ongelmatilanteet ja niiden korjaamiset. ETP:n tapauksessa käynnistämisen yhteydessä tarkastetaan esimerkiksi transparent-reunukset, jotka piilotetaan tietyn väriskaalauksen mukaan. Kuvassa 17 esiteltynä ohjelmakehyksien transparent-reunukset päällä- sekä pois-tilanteissa.



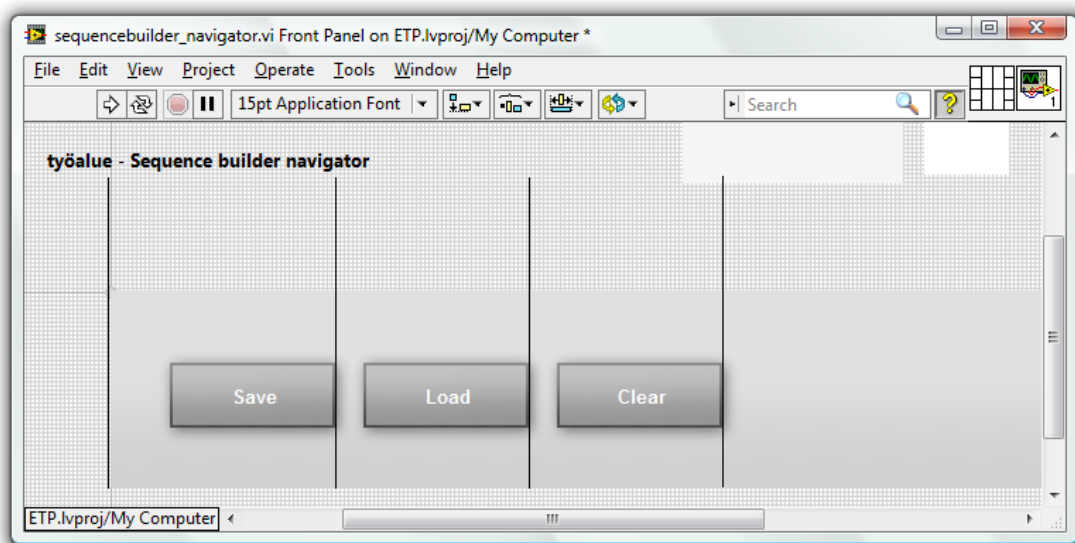
Kuva 17. ETP:n ohjelmakehyksien tulostus.

Otsikkopalkilla tarkoitetaan sitä aluetta, josta voidaan siirtää ohjelmisto käyttäjän mieltymysten mukaisesti. Otsikkopalkissa sijaitsevat ohjelman päänapulat, jotka ovat ohjelman sammutus, pienennys ja avustus. ETP:n otsikkopalkin testaukseen kuului näiden nappuloiden toimivuus aina, kun niiden pitääkin sekä niiden estäminen tarpeiden mukaisesti. Yksi testauksen osista oli myös ikkunan siirtely, painamalla hiirtä pohjassa ja siirtämällä ikkunaa haluttuun sijaintiin ruudussa.

Navigaattorilla tarkoitetaan puunäkymää, jolla käyttäjä voi selata ohjelmiston eri osia tarpeiden mukaisesti. Navigaattorissa sijaitsee ohjelmiston kaikki päätyökalut, joita käyttäjä tarvitsee saadakseen työnsä tehdyksi. Testaukseen

kuului puunäkymän toiminnan takaaminen. Tällä varmistettiin, että ohjelmisto suorittaa valitun työkalun tulostamisen ruudulle aina kun se on valittuna.

Hallintapaneelilla tarkoitetaan alhaalla olevaa elementtiä, millä käyttäjä voi hallita sen hetkistä näkyvillä olevaa työaluetta. Hallintapaneelissa vaihtuu kontrollit, riippuen mitä työkalua tai ominaisuutta käyttäjä sillä hetkellä käyttää. Hallintapaneelin testauksessa lähinnä tarkistettiin, että kaikki tarvittavat kontrollit tulivat näkyviin ja ne tekivät niille tarkoitettut tehtävät. Tällainen on esimerkiksi Save-nappula, jonka tarkoitus on tallentaa sen hetkiset asetukset, jotka ovat työalueella näkyvissä. Testauksessa myös tarkistettiin kontrollien kohdistus, jolloin kaikki nappulat olivat oikeissa kohdissa, työkalusta riippumatta. Kuvassa 18 hallintapaneelin kontrollien kohdistamisesta.

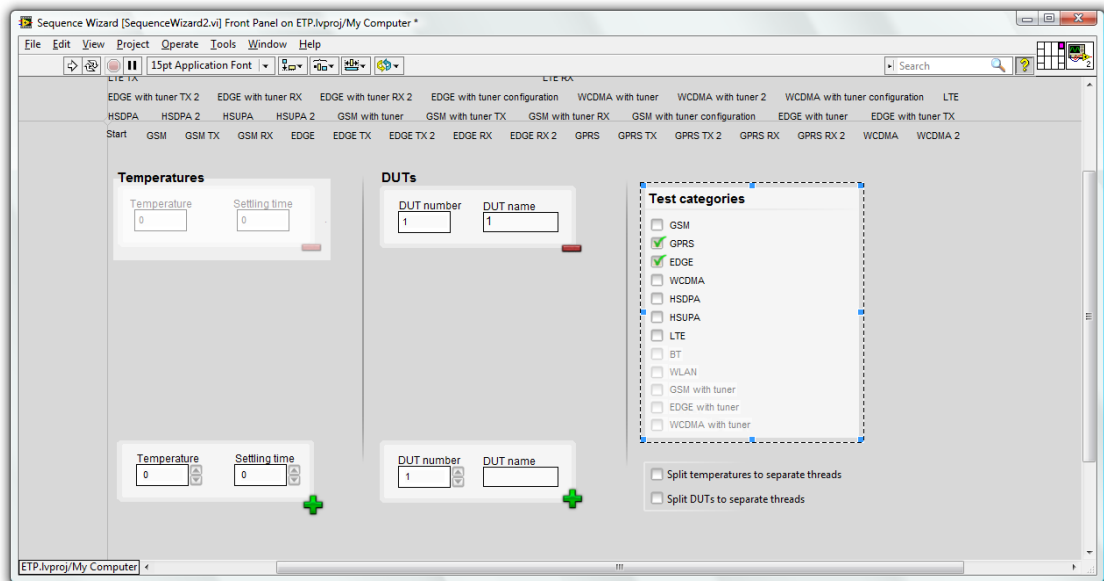


Kuva 18. Hallintapaneelin kontrollien kohdistus.

Työalueella tarkoitetaan aluetta, jossa näkyy sen hetkisen työkalun mahdolliset toimenpiteet tai asetukset. Työalueen toimivuus testattiin käymällä eri työkaluja läpi niin, että ne tulostuivat työalueella täysin oikein. Työalueen testauksiin myös kuului erilaisten elementtien kohdistaminen, niin että kaikki

samantyyppiset asetukset tai kontrollit olivat samassa kohtaa ruudulla, työkalusta riippumatta.

ETP:n työkalujen testauksessa käytettiin tekniikkaa, jossa työkalun valmiiksi tulemisen jälkeen se testattiin mahdollisimman nopeasti. Työkalun testaamiseen käytettiin mahdollisimman monta henkilöä, josta saatiin tehokkaasti isoimmat ongelmat korjattua erilleen. Tällainen työkalu on mm. sekvenssin rakentamiseen tarkoitettu avustajatyökalu, joka on monisyinen prosessi, jossa edetään askel askeleelta riippuen käyttäjän asetuksista. Tällaisen prosessin testaus vaatii aikaa ja vaivaa, jotta suurimmat virheet löydettiin ja näin ollen korjattiin. Kuvassa 19 sekvenssin rakentamiseen tarkoitettu työkalun esittely testausvaiheessa.



Kuva 19. Sekvenssien rakennus työkalun testausta.

7 YHTEENVETO

Tässä työssä on käsitelty käyttöliittymän luomista ja siihen liittyviä tärkeitä asioita sekä ongelmia. Hyvän käyttöliittymän toteutus lähtee aina liikkeelle suunnittelusta ja selkeästä kokonaisuudesta. Käyttöliittymä pitäisi aina rakentaa käyttäjälle helposti käytettäväksi ja nopeasti opittavaksi, joka monesti jää unholaan monimutkaisten ominaisuuksien ja turhien painalluksien päähän.

ETP:n käyttöliittymä toteutus onnistui projektina hyvin. Aikataulussa onnistuttiin pysymään ja ilman suurempia ongelmia vältyttiin. Työstä saatiin myös ensimmäiset suunnitteluviedokset jo nopeasti julkastu, mikä toi lisää intoa itse työhön. Testaukseen olisi voinut panostaa hieman enemmän, jotta erilaisia näkökulmia olisi syntynyt ja näin ollen saatu vieläkin pidemmälle vietyjä työkaluja aikaiseksi. Testauksen yhteydessä oli hienoa myös huomata, miten aikaisemmin ETP:tä käyttäneet eivät huomanneet kaikkia perusominaisuuksia, jotka taas tulivat kahdelta muulta henkilöltä aivan ensimmäisiksi mieleen. Tällainen oli mm. TAB -näppäimen käyttö, jossa käyttäjä voi loikkia nopeasti eri tekstikenttiä näppäimistön avulla.

Suurimpina toteutusongelmina oli mahdollisesti LabView-ohjelmiston käyttäminen. Monia asioita, joiden luomiseen olisi luullut menevän vain vähän aikaa, menikin yllättävän paljon. Hyvänä esimerkkinä kontrolleiden luonti, jossa LabView:ssä oli omat työkalunsa näitä varten. Työkalut olivat kuitenkin hieman vajanaiset ja muutamia ominaisuuksia kärsi alkuperäisestä suunnitelmasta. Tällaisia ominaisuuksia oli mm. listoille tarkoitetut muotoilut, joita ei voitu toteuttaa halutulla tavalla.

Projektin tavoitteet saatiin saavutettua aikataulun mukaisesti ja ohjelmisto on näin ollen jo käytettävissä hienosäädöistä huolimatta. Suurimmat ongelmatkin

saatiin ratkaistua. Graafinen ulkomuoto päivitettiin ajantasalle, joka oli suurin ongelma lähtötilanteessa ja käytettävyyden tehostaminen uuden puurakenteen ansioista, jolla saatiin käytettävyys nopeammaksi ja helpommaksi.

LÄHTEET

[1] [www-dokumentti]. Artem, P., Olio-ohjelmointi LabView-ympäristössä, opinnäytetyö, Metropolia Ammattikorkeakoulu, 2011, viitattu 20.4.2013 <http://publications.theseus.fi/bitstream/handle/10024/48441/Thesis%20Artem%20Pirojenko%20print.pdf?sequence=1>

[2] [www-dokumentti]. LabView ohjelmointi 1996, viitattu 20.4.2013 <http://www.mit.jyu.fi/opiskelu/seminaarit/bak/labview/>

[3] [www-dokumentti]. The Human Interface 2009, viitattu 20.4.2013 <http://nitpicker.pbworks.com/w/page/12451253/The%20Humane%20Interface>

[4] [www-dokumentti]. Wireframes 2013, viitattu 20.4.2013 <http://konigi.com/node/1819>

[5] [www-dokumentti]. The what, when and why of wireframes 2009, viitattu 20.4.2013 <http://userpathways.com/2008/06/the-what-when-and-why-of-wireframes/>

[6] [www-dokumentti]. Iivonen, S., Käyttöliittymän Suunnitteluprosessi, opinnäytetyö, Lahden ammattikorkeakoulu, 2012, viitattu 20.4.2013 http://publications.theseus.fi/bitstream/handle/10024/42977/iivonen_Sami-Petteri.pdf?sequence=1

[7] [www-dokumentti]. Käyttöliittymät ja käytettävyys 2004, viitattu 20.4.2013 <http://web.archive.org/web/20091020041038/http://www.adage.fi/blogi/2004/kayttoliittymat-ja-kaytettavyys/>

[8] [www-dokumentti]. 10 Usability Heuristics for User Interface Design 2005, viitattu 20.4.2013 <http://www.nngroup.com/articles/ten-usability-heuristics/>

- [9] [www-dokumentti]. Graafinen käyttöliittymä 2012, viitattu 20.4.2013
http://fi.wikipedia.org/wiki/Graafinen_k%C3%A4ytt%C3%B6liittym%C3%A4
- [10] [www-dokumentti]. Sonninen, K., Graafisen käyttöliittymän suunnittelu käytettävyyden näkökulmasta, insinöörityö 2011, viitattu 20.4.2013
http://publications.theseus.fi/bitstream/handle/10024/38032/Sonninen_Kalle.pdf?sequence=1
- [11] [www-dokumentti]. Värit ja niiden merkitys käyttöliittymän visuaalisessa suunnittelussa 2003, viitattu 20.4.2013 http://www.soberit.hut.fi/T-121/T-121.200/suomi/syksy2003/essee2003/mari_mankki.pdf
- [12] [www-dokumentti]. Käyttöliittymäntestaus 2006, viitattu 20.4.2013
<http://www.oamk.fi/sbc/testaus/kayttoliittymatestaus.htm>