



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Antti Mikael Pöntinen

SÄHKÖNJAKELUVERKOSTA
KERÄTTYJEN
VIKATAPAHTUMATALLENTEIDEN
ESITYSTYÖKALU

Tekniikka ja liikenne
2010

ALKUSANAT

Tämä opinnäytetyö tehtiin Vaasan ammattikorkeakoulun tietotekniikan koulutusohjelman päättötyönä ABB Oy Sähkönjakeluautomaatiolle. Työn tarkoituksena oli kehittää työkalu, jolla IED:n häiriötilanteessa tallentamia eri työkaluista kerättyjen tietojen perusteella tehtyjä vikatapahtumatallenteita olisi mahdollista sekä esittää selkeässä muodossa kootusti yhdessä näkymässä, että tallentaa käyttäjän omalle työasemalle.

Työn valvojana toimi Vaasan ammattikorkeakoulun tekniikan ja liikenteen yksiköstä yliopettaja Pirjo Prosi. Ohjaajana sekä yhdyshenkilönä toimi ABB Oy Sähkönjakeluautomaation SW Products -osaston tuotekehityspäällikkö Ville Luoma.

Haluan kiittää tuotekehityspäällikkö Ville Luomaa valitsemastaan aiheesta ja antamastaan mahdollisuudesta tehdä tämä opinnäytetyö ABB Oy Sähkönjakeluautomaatiolle. Haluan myös kiittää tuotekehityksen työntekijöitä ja erityisesti suunnitteluarkkitehti Henry Niveriä, jotka tarjosivat arvokasta apuaan työn eri vaiheissa, sekä yliopettaja Pirjo Prosia työn valvojana toimimisesta.

Vaasassa 4.6.2010

Antti Pöntinen

VAASAN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma

TIIVISTELMÄ

Tekijä	Antti Pöntinen
Opinnäytetyön nimi	Sähkönjakeluverkosta kerättyjen vikatapahtumatallenteiden esitystyökalu
Vuosi	2010
Kieli	Suomi
Sivumäärä	43 + 1 liitettä
Ohjaaja	Pirjo Prosi

Tämä opinnäytetyö tehtiin Vaasan ammattikorkeakoulun tietotekniikan koulutusohjelman päättötyönä ABB Oy Sähkönjakeluautomaatiolle. Työn tarkoituksena oli kehittää vikatapahtumatallenteiden analysointia helpottava työkalu, jolla pystyttäisiin tarkastelemaan ja tallentamaan IED:n tuottamia vikatapahtumatallenteita kootusti selkeässä muodossa.

Työ aloitettiin ohjelmoimalla Microsoft Office Excel -taulukkolaskentaohjelmassa ajettava makro, jolla luotiin käytössä olleesta taulukosta XML-tiedosto. Tämä tiedosto sisälsi attribuutteina vikatapahtumatallenteiden parametrien nimet, lyhenteet ja yksiköt. Vikatapahtumatallenteiden sisältämät parametrit riippuvat käytössä olevasta IED:stä ja sen konfiguraatiosta, joten tästä tiedostosta oli itse sovelluksessa helpompi lukea tarvittavien parametrien nimet, lyhenteet ja yksiköt.

Työkalun toteutus aloitettiin niin kutsutulla protoiluvaiheella, jossa tehtiin esimerkkejä käyttöliittymästä, joiden pohjalta saatujen palautteiden ja toivomusten mukaan tehtiin uusia malleja. Samalla määriteltiin sovelluksen vaatimukset. Toiminnallisuudet lisättiin myöhemmin ja niitä testattiin jatkuvasti. Hitaiden testien ja korjausehdotusten takia toteutusta ei ehditty viedä niin pitkälle kuin alunperin oltiin suunniteltu. Sovelluksen toteuttamiseen käytettiin Microsoftin kehittämää oliopohjaista C#-ohjelmointikieltä Microsoft .NET-ympäristössä. Sovelluskehityksessä käytettiin Microsoft Visual Studio 2008 -kehitysympäristöä.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietotekniikan koulutusohjelma

ABSTRACT

Author	Antti Pöntinen
Title	Tool for Displaying the Fault Records Collected from the Electricity Distribution Network
Year	2010
Language	Finnish
Pages	43 + 1 Appendices
Name of Supervisor	Pirjo Prosi

This thesis was conducted for ABB Oy Distribution Automation. The aim of this thesis was to develop a tool to ease the analyzing of the Fault Records. It was necessary to be able to read the Fault Records from the IED, show them on a graphical user interface and save them to a text file.

Work was started by programming a macro for the Microsoft Office Excel. This macro collects all the parameter names and their abbreviations from the spreadsheet in use and saves them to an XML-file. This file is then read into the tool itself. The content of the Fault Record depends on what IED and its configuration is in use, so the macro is meant to make it easier to collect all the necessary information from the spreadsheet to be used in the tool.

Implementation of the tool was started by so called proto phase, where different user interface models were made according to the feedback and proposals from the previous models. At the same time requirements for the tool were made. After that, the functionality itself was added and tested during the implementation. Because of slow tests and fix proposals, some of the planned features had to be left out. The application was programmed using C# and Microsoft .NET. Microsoft Visual Studio 2008 was used as a development environment.

Keywords C#, Excel, distribution of electricity, XML

SISÄLLYS

ALKUSANAT	2
TIIVISTELMÄ.....	3
ABSTRACT	4
KÄYTETYT MERKINNÄT JA LYHENTEET	7
1 JOHDANTO	9
1.1 Opinnäytetyön taustat ja tavoitteet	9
1.2 ABB Oy Sähkönjakeluautomaatio.....	9
2 SÄHKÖNJAKELU	13
2.1 Yleistä	13
2.2 Keskijänniteverkot ja sähköasemat.....	13
3 RELESUOJAUS	15
3.1 Yleistä	15
3.2 Numeeriset releet	15
4 VIKATAPAHTUMATALLENTEET	17
4.1 Yleistä	17
4.2 IEEE COMTRADE -standardi.....	20
4.2.1 Yleistä	20
4.2.2 Tiedostot.....	21
5 TYÖKALU	26
5.1 Ohjelmistotuotannon vaiheet.....	26
5.2 Määrittely	27
5.3 Suunnittelu	28
5.3.1 Yleistä	28
5.3.2 Arkkitehtuurisuunnittelu	29
5.4 Toteutus.....	31
5.4.1 Yleistä	31
5.4.2 Excel makro.....	31
5.4.3 ConnPack	33
5.4.4 Fault Record Tool.....	34
5.5 Testaus	38

6	YHTEENVETO.....	40
	LÄHDELUETTELO.....	41
	LIITELUETTELO.....	43

KÄYTETYT MERKINNÄT JA LYHENTEET

ANSI	American National Standards Institute, valvoo standardien kehittymistä Yhdysvalloissa
ASCII	American Standard Code for Information Interchange, tietokoneiden merkistö
COMTRADE	Common Format for Transient Data Exchange. IEEE -standardi, joka määrittää yleisen ulkoasun esimerkiksi häiriödatatiedostoille
ConnPack	Connectivity Package, liitettävyyspaketti. Kokoelma ohjelmia joilla liitetään IED:hin
IED	Intelligent Electronic Device, mikroprosessoripohjainen kontrolleri
IEEE	Institute of Electrical and Electronics Engineers, Inc, kansainvälinen tekniikan alan järjestö
ISO	International Organization for Standardization, kansainvälinen standardisointijärjestö
ISO 9001	Tuotekehitystä ja tuotantoa sisältävään yritykseen tarkoitettu laatustandardi. Osa laatujärjestelmiin vaikuttavaa ISO 9000 -standardisarjaa
.NET	Microsoft .NET Framework, Microsoftin kehittämä ohjelmistokomponentti kirjasto
PCM	Protection and Control IED Manager, IED managerointityökalu
VBA	Visual Basic for Applications, Visual Basic -ohjelmointikielen Microsoft Office -ohjelmiin sisäänrakennettu toteutus
VBE	Visual Basic Editor, Microsoft Office -ohjelmiin sisäänrakennettu Visual Basic -ohjelmointikielellä käytettävä ympäristö

XML Extensible Markup Language, rakenteellinen kuvauskieli
tiedonvälitykseen järjestelmien välillä

1 JOHDANTO

1.1 Opinnäytetyön taustat ja tavoitteet

Kun sähköverkossa tapahtuu jokin vikatilanne, tallentaa mikroprosessoripohjainen suojarile eli IED (Intelligent Electronic Device) eri työkalujen keräämien tietojen perusteella tapahtuneesta virheestä vikatapahtumatallenteen. Tällä hetkellä tallenne on kuitenkin vaikeasti luettavassa muodossa. Tämän opinnäytetyön tarkoituksena on kehittää tilannetta helpottava työkalu, jolla pystytään esittämään IED:n tuottama vikatapahtumatallenne selkeässä muodossa, sekä keräämään eri työkalujen tuottamat tiedot yhteen ja samaan näkymään. Lisäksi työkalulla tulisi olla mahdollista tallentaa tallenne halutussa tiedostomuodossa käyttäjän omalle työasemalle.

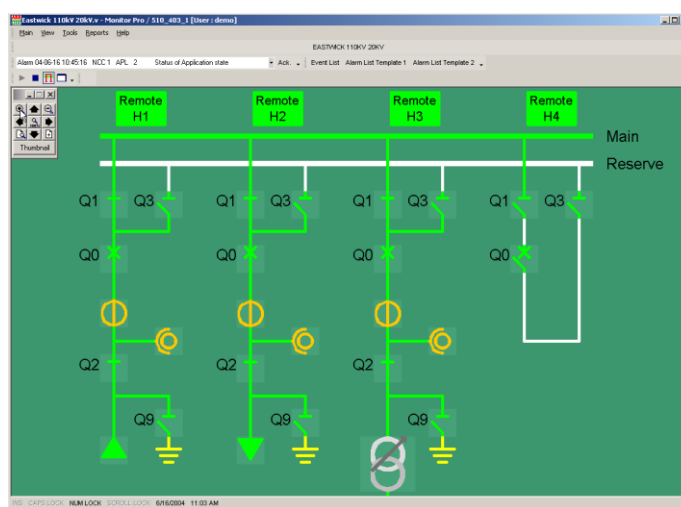
Lähtökohtana alusta asti on ollut vaiheittainen toteutustapa. Tarkoituksena on ensimmäisenä toteuttaa perustoiminnoilla varustettu työkalu, joka tuo apua tärkeimpiin ongelmakohtiin. Eri lisäominaisuuksia tullaan myöhemmin lisäämään tarpeen mukaan. Työkalua tullaan jatkokehittämään opinnäytetyön jälkeenkkin, joten tämä opinnäytetyö käsittelee työkalun kehitystä ja sen ominaisuuksia niiltä osin kun niitä ajan puitteissa saadaan valmiiksi.

IED:llä tarkoitetaan tässä yhteydessä pääasiassa ABB:n RE_615-sarjan mikroprosessoripohjaista suojarilettä, mutta se voi tarkoittaa yleisesti ottaen mitä tahansa, yleensä sähköteollisuudessa käytettyä mikroprosessoripohjaista kontrolleria, joka pystyy vastaanottamaan erilaisia ohjauskäskyjä. Nykypäivänä valtaosa suojarileistä ovat mikroprosessoripohjaisia, koska sen ansiosta erilaisia sähköverkon suojaus- ja ohjaustoimintoja pystytään yhdistämään yhteen ja samaan laitteeseen. /7/

1.2 ABB Oy Sähkönjakeluautomaatio

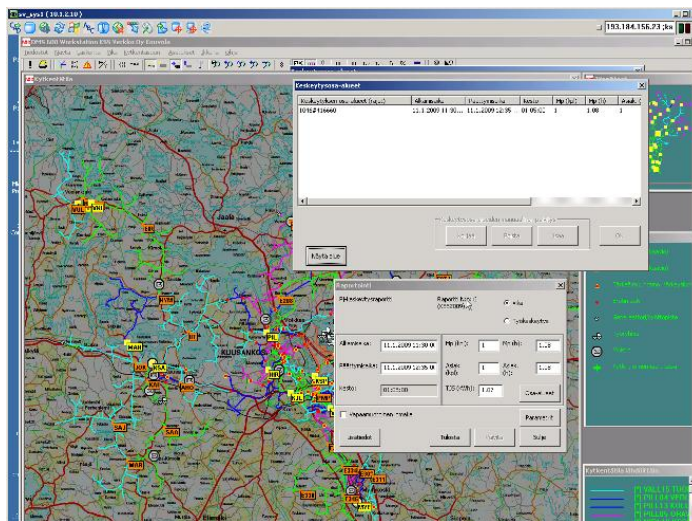
ABB Oy Sähkönjakeluautomaatio on globaali markkinajohtaja suojarileiden myynnissä ja sen tarkoituksena on turvata sähkölaitos- ja teollisuusasiakkaillensa häiriötön sähkönjakelu. Yrityksen liikevaihto vuonna 2008 oli 92 miljoonaa

euroa. Sähkönjakeluautomaatiolla tarkoitetaan keskijänniteverkon sähkönjakelun suojausta, valvontaa, ohjausta ja optimointia. ABB Oy Sähkönjakeluautomaation tuotteita ovat erilaiset sähköverkon suojarieleet, sekä sähköverkon hallintajärjestelmät, joita ovat MicroSCADA Pro -käytönvalvontajärjestelmä, DMS 600 -käytöntukijärjestelmä, sekä Asema-automaatio COM 600. Lisäksi yritys tarjoaa erilaisia palvelutuotteita, kuten teknistä tukea, ennakkohuoltoa sekä asiakaskoulutuspalveluita. Alla olevissa kuvissa näkyy kunkin hallintajärjestelmän käyttöliittymät.



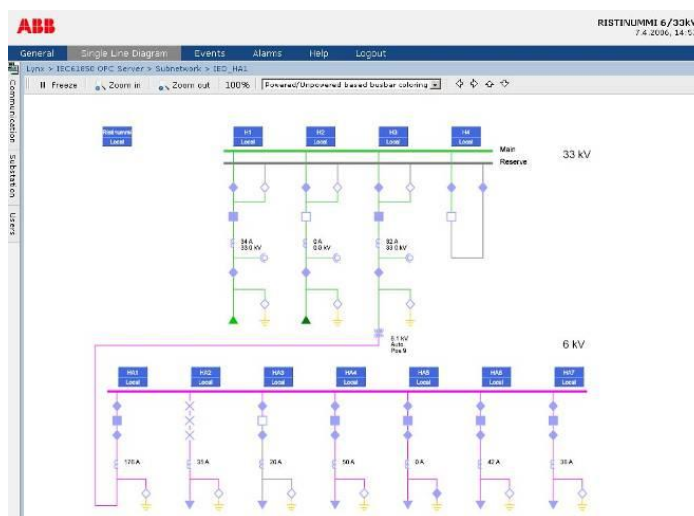
Kuva 1. MicroSCADA Pro -käytönvalvontajärjestelmä. /4/

MicroSCADA Pro kertoo käyttäjälle järjestelmän tilasta ja sillä on mahdollista ohjata sähkönjakelujärjestelmää, kuten suorittaa vianpaikannuksia ja jännitteenpalautuksia. Liitäntä sähkönjakeluverkkoon tapahtuu suojarieiden kautta.



Kuva 2. DMS 600 -käyttöjärjestelmä. /4/

DMS 600:n tarkoituksena on minimoida sähkökatkosten kestoja mm. reaaliaikaisen tilaseurannan ja taustakarttojen avulla. DMS 600 on integroitu MicroSCADA:an.



Kuva 3. Asema-automaatio COM 600. /4/

COM 600 sijaitsee yleensä sähköasemalla ja sisältää yhdyskäytävätoiminnot sähköaseman kaikkien IED:den ja sähköverkon hallintajärjestelmien välillä.

Sähkönjakeluautomaation rooli ABB-yhtymässä on maailmanlaajuinen vastuu sähkönjakeluautomaation suojaruleiden, työkaluohjelmistojen ja

kommunikaatiolaitteiden kehityksestä, markkinoinnista, myynnistä, asiakastuesta ja tuotannosta. Yrityksellä on myös vastuu tulevaisuuden teknologioiden ja ratkaisujen kehittämisestä.

Sähköjakeluautomaation henkilökunta on hyvin kansainvälinen ja toimihenkilövaltainen. Henkilöstöä organisaatiossa on 298, joista toimihenkilöitä on 92% ja tuotannon työntekijöitä on 8%. Tuotekehityksen osuus koko henkilöstöstä on 35%. Henkilöstöä on 12:sta eri maasta ja jatkuvaa työntekijävaihtoa tehdään eri maiden välillä. Tuotteista yli 90% menee vientiin ja myyntiverkosto toimii yli 70:ssä maassa. /4/

2 SÄHKÖNJAKELU

2.1 Yleistä

Sähkönjakelulla tarkoitetaan sähkövoimansiirtojärjestelmän kautta tulevan tai jakeluverkkoon liitettyjen voimalaitosten tuottaman sähkön siirtoa loppukäyttäjille. Järjestelmän osiin kuuluu alueverkko (110kV ja 45kV), sähköasemat (110/20kV, 45/20kV), keskijänniteverkko (20kV), jakelumuuntamot (20/0,4kV) sekä pienjänniteverkko (0,4kV).

Primäärikomponenttien lisäksi sähkönjakelujärjestelmään kuuluu sekundäärilaitteita ja -järjestelmiä. Näistä ABB Oy Sähkönjakeluautomaatio tuottaa ja kehittää keskijänniteverkon sähköasemilla olevia suojarkeitä, sekä käyttökeskuksissa käytössä olevia käytönvalvonta- ja käytöntukijärjestelmiä. /8/

2.2 Keskijänniteverkot ja sähköasemat

Keskijänniteverkolla tarkoitetaan sähköverkkoa, jonka jännitetaso on joissakin kaupungeissa käytössä oleva 10kV, tai yleisemmässä käytössä oleva 20kV. Keskijänniteverkko on hyvin tärkeä osa sähköverkkoa, kun puhutaan sen toimintavarmuudesta. Keskijänniteverkossa tapahtuvien vikojen osuus koko verkossa tapahtuvista vioista on jopa yli 90%. Lisäksi keskijänniteverkko toimii varayhteytenä pahoissa sähköasemien ja 110kV:n johtojen vikatilanteissa.

Sähköasema on tärkein yksittäinen osa sähkönjakeluverkossa. Sen tarkoituksena on muuntaa jännitetaso sopivaksi sen syöttämään sähköverkkoon. Keskijänniteverkon kohdalla sähköasema muuntaa suuremman, yleensä 110kV:n jännitteen joko 10kV:n tai 20kV:n tasolle. Sähköasemaan kuuluu suurjännitekytkinlaitos, ainakin yksi päämuuntaja, keskijännitekytkinlaitos ja apujännitejärjestelmä käytöntukitoimintoinen. Näistä muuntaja on sähköaseman kallein komponentti.

Sähköasemalla sijaitsee pääosa sähköverkon suojarkeitä ja automaatiosta. Päämuuntajan pääsuojina toimivat ylivirtarele ja differentiaalirele, joista

differentiaalireleen tarkoitus on muuntajan vikojen tunnistus, joita ovat oikosulku, maasulku, käämisulku ja kierrossulku. /8/

3 RELESUOJAUS

3.1 Yleistä

Suojarele on laite, jonka tehtävä on suojata sähköverkkoon kytkettyjä laitteita ja järjestelmiä vaurioilta, sekä ehkäistä henkilövahinkoja. Kun sähköverkossa tapahtuu suojareleeseen asetellun toiminta-arvon ylittävä muutos, antaa se käskyn viallisen kohteen kytkemiseksi pois järjestelmästä. Nykyaikaiset suojareleet ovat lisäksi liitettyinä käytönvalvonta- ja käytöntukijärjestelmiin, joiden tarkoituksena on nopeuttaa vikojen havaitsemista sekä korjaamista. Nykyaikaisilla suojareleilla on myös oheistoimintoja, jotka auttavat varmistamaan sähköasemien ja sähköjärjestelmien tehokkaan toiminnan. /4/, /9/

3.2 Numeeriset releet

Numeerisille releille eli prosessorireleille on tunnusomaista, että suojauskohteen erilaiset suojaustoiminnot ovat integroituna samaan releeseen. Nykyisin niistä pystytään lukemaan mittaus-, tila- ja asetteluarvoja, sekä suojarele kykenee vastaanottamaan ohjaus- ja asettelutietoja. Tästä tiedunkeruusta on hyötyä jälkikäteen tehtävässä vikojen analysoinnissa. Kuvassa 4 on ABB Oy:n Relion-tuoteperheeseen ja sen 615-tuotesarjaan kuuluva REF615 IED. /9/



Kuva 4. REF615 IED. /1/

4 VIKATAPAHTUMATALLENTEET

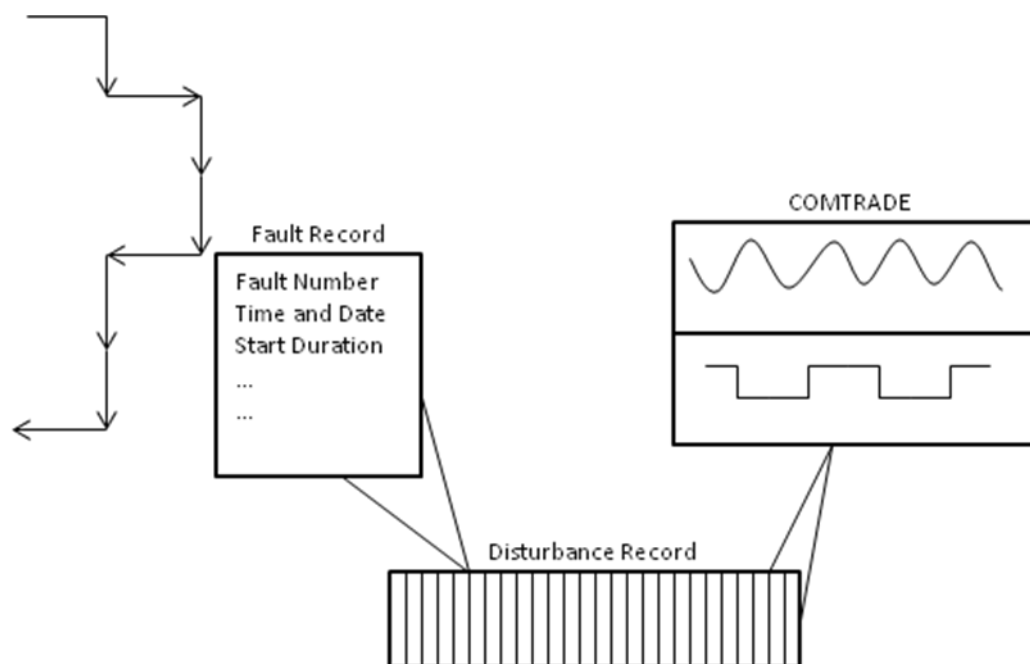
4.1 Yleistä

Vikatapahtumatallenne on vapaa suomennos termistä Fault Record, jota ei kuitenkaan pidä sekoittaa häiriötallenteeseen, joka tulee termistä Disturbance Record. Termien merkityksestä ja eroista kerrotaan seuraavaksi.

IED:n mukana tulee häiriötallennin, joka sisältää 12 analogista ja 64 digitaalista kanavaa. Analogikanavat voidaan asettaa laukaisemaan tallennustoiminto jos mitattava arvo alittaa tai ylittää asetetun arvon. Tallennus voidaan laukaista myös manuaalisesti, tietyin aikaväleihin tai digitaalikanavien kohdalla tilamuutosten perusteella. Päällä ollessaan IED suorittaa jatkuvasti sarjoja komentoja, jotka pitävät yllä tietyn määritetyn ajanjakson mukaista tilastoa sähköverkon vallitsevista virta- ja jännitearvoista sekä sen saamista ohjaus- ja hallintakäskyistä. Mahdollisen sähköverkossa tapahtuvan vikatilanteen tapahtuessa tallennetaan tästä tilastosta halutun puskurin kokoinen häiriötallenne (Disturbance Record), joka voi sisältää analogikanavien tallentamat arvot sen hetkisistä virta- ja jännitearvoista, sekä digitaalikanavien tallentamat sisäiset ja ulkoiset signaalit, kuten esimerkiksi hallintakäskyt. Häiriötallenteet tallennetaan COMTRADE -standardin mukaisesti data- ja konfiguraatitiedostoon IED:n muistiin. /2/

Häiriötallenne pitää sisällään useita vikatapahtumatallenteita (Fault Record). Tarkemmin ilmaistuna häiriötallenne koostuu vikatapahtumatallenteista, joita tallennetaan tietyin aikaväleihin nopeasti luettavalle, niin kutsutulle haihtumattomalle muistille, joka pystyy säilyttämään siihen tallennetut tiedot ilman sähköäkin. Näiden vikatapahtumatallenteiden sisältämät tiedot riippuvat kyseessä olevasta IED:stä, sekä sen käytetystä konfiguraatiosta. Kaikki tallenteet sisältävät kuitenkin vähintään tallenteen numeron, aikaleiman ja vapaasti suomennettuna aloituskeston. Tämä aloituskesto kertoo haluttaessa esimerkiksi prosentteina, missä kohtaa niin kutsuttujen start- ja trip-rajojen välillä vikatapahtumatallenne on otettu. Eli jos liipaisu on tapahtunut esimerkiksi puolivälissä start- ja trip-rajaa, on aloituskesto silloin 50%. Kuvassa 5 on havainnollistettu vikatapahtumatallenteiden ja häiriötallenteiden syntyä ja niiden

tallennusta. Taulukossa 1 on esitettyä osa tiedoista, joita vikatapahtumatallenne voi pitää sisällään. /10/



Kuva 5. Kuvaus vikatapahtumatallenteiden ja häiriötallenteiden synnystä ja tallennuksesta. /10/

Taulukko 1. Vikatapahtumatallenteen data /2/

Name	Type	Values(Range)	Unit	Description
Number	INT32	0...999999		Fault record number
Time	Timestamp			Time of recording
LNPLDF1 duration	FLOAT32	0.00...100.00	%	LNPLDF1 Start duration
Max current IL1	FLOAT32	0.000...50.000	xln	Maximum phase A current
Max current IL2	FLOAT32	0.000...50.000	xln	Maximum phase B current
Max current IL3	FLOAT32	0.000...50.000	xln	Maximum phase C current
Max current IO	FLOAT32	0.000...50.000	xln	Maximum residual current
Max diff current IL1	FLOAT32	0.000...80.000		Maximum phase A differential current
Max diff current IL2	FLOAT32	0.000...80.000		Maximum phase B differential current
Max diff current IL3	FLOAT32	0.000...80.000		Maximum phase C differential current
Max bias current IL1	FLOAT32	0.000...50.000		Maximum phase A bias current
Max bias current IL2	FLOAT32	0.000...50.000		Maximum phase B bias current
Max bias current IL3	FLOAT32	0.000...50.000		Maximum phase C bias current
Voltage UL1	FLOAT32	0.000...4.000	xUn	Phase A voltage
Voltage UL2	FLOAT32	0.000...4.000	xUn	Phase B voltage
Voltage UL3	FLOAT32	0.000...4.000	xUn	Phase C voltage
Angle U23 - IL1	FLOAT32	-180.00...180.00	deg	Angle phase B to phase C voltage - phase A current
Angle U31 - IL2	FLOAT32	-180.00...180.00	deg	Angle phase C to phase A voltage - phase B current

Angle U12 - IL3	FLOAT32	-180.00...180.00	deg	Angle phase A to phase B voltage - phase C current
PTTR thermal level	FLOAT32	0.00...99.99		PTTR calculated temperature of the protected object relative to the operate level

Vanhemmissa RE_615-sarjan IED:n versioissa vikatapahtumatallenteita pystyttiin tallentamaan kerrallaan vain neljä kappaletta. Nykyisissä versioissa niitä pystytään tallentamaan kerrallaan 32 kappaletta ja tulevaisuudessa versioissa todennäköisesti vielä enemmän. Tästä johtuen vikatapahtumatallenteiden helpommasta analysoinnista tulee tärkeä osa IED:n käytettävyyden parantamisessa. Näiden tallennettujen tiedostojen rakenteet ja sisältö ovat tarkasti määritelty IEEE COMTRADE -standardissa, jota ABB Oy:n IED:t käyttävät.

4.2 IEEE COMTRADE -standardi

4.2.1 Yleistä

Digitaalisen teknologian kasvava käyttö erilaisissa suoja-, mittaus- ja hallintalaitteissa sähköjakelukeskuksissa on luonut mahdollisuuden kerätä suuria määriä digitaalisia tallenteita sähköverkon hetkellisistä tapahtumista. Lisäksi erilaisia analogisia ja digitaalisia sähköverkon simulaattoreita voidaan käyttää digitaalisten tallenteiden luonnissa. Tätä dataa käytetään erilaisten laitteiden kanssa sähköverkon ja sen suojauksen analysoinnin, testauksen, arvioinnin ja simuloinnin automatisoinnin parantamisessa vika- ja häiriötilanteissa. Näiden tallenteiden käytössä kohdataan ongelmia, kun joudutaan tulemaan toimeen eri lähteiden erilaisten patentoitujen muotojen kanssa, joita eri järjestelmät käyttävät tietojen luontiin, tallennukseen ja lähettämiseen eri sovellusten välillä.

IEEE COMTRADE -standardin tarkoituksena on luoda yleinen muoto datatiedostoille ja vaihtovälineille, joita käytetään sähköverkon eri tyyppisten häiriö-, testi- tai simulaatiodatoiden korvaamiseen. Hetkellisen datan lähteet on kuvattu, näytteenottotaajuudet, suodattimet ja näytteenottotaajuuksien

muunnokset hetkelliselle datalle on käyty läpi, sekä datan vaihtoon tarkoitettut tiedostot ja niiden rakenteet on määritelty.

Standardista on alkuperäisen IEEE Std C37.111-1991 -version lisäksi IEEE Std C37.111-1999 -versio, sekä vuoden 2005 revisio. /6/

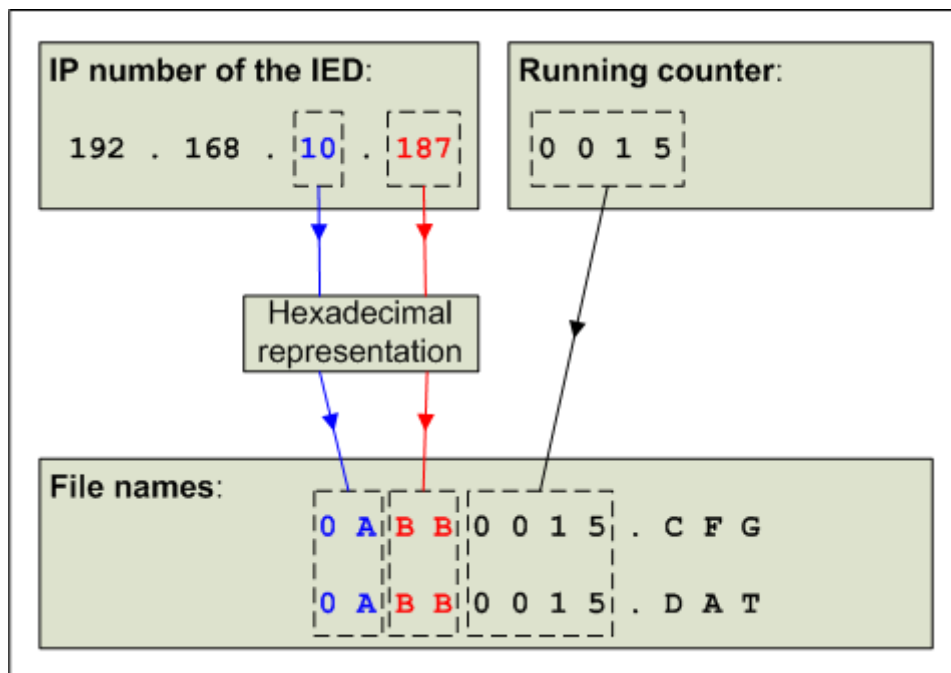
4.2.2 Tiedostot

On olemassa kolme yleistä tiedostoluokkaa, joita tietokoneet käyttävät: suoritettavat tiedostot, tekstitiedostot ja datatiedostot. COMTRADE -standardissa määritellään teksti- ja datatiedostot, mutta se ei määrittele suoritettavia tiedostoja ollenkaan. /6/

COMTRADE -standardin mukaiset tekstitiedostot käyttävät ANSI X3.4-1986 mukaista merkkien esitystä. Tämä tunnetaan paremmin ASCII- tai .txt-tyyppisenä esityksenä. COMTRADE määrittelee yhden, yksinomaan ihmisille tarkoitetun ASCII-tyyppisen tekstitiedoston, sekä kolme muuta sekä ihmisille että tietokoneille tarkoitettua ASCII-tyyppistä tekstitiedostoa. Käyttämällä ASCII-tyyppisiä tiedostoja pystytään välttymään erilaisilta tekstinkäsittelyohjelmien riippuvuuksilta. /6/

Datatyypiset tiedostot voivat pitää sisällään numeerista dataa, tekstipohjaista dataa, tai molempia ja ne voidaan tallentaa joko binääri- tai ASCII-tiedostoina. Binääristä dataa joudutaan erikseen prosessoimaan että se saadaan luettavaan muotoon. Sen etu tulee kuitenkin esille, jos dataa on suuria määriä, koska se pystytään tallentamaan pieneen tilaan. COMTRADE määrittelee yhden binääritiedoston. /6/

COMTRADE -standardin mukainen tallenne voi sisältää neljä eri tyyppistä tiedostoa, joita ovat otsikko-, konfiguraatio-, data- sekä informaatiotiedosto. Jokaisen tiedoston nimen täytyy olla täsmälleen sama, vain tiedostopääte erottaa tiedostot toisistaan. Tiedostonimen maksimi pituus on kahdeksan merkkiä, kun tiedostopäätteellä se on kolme merkkiä. Seuraavissa kappaleissa käydään kaikki eri tyyppiset tiedostot läpi. Kuvassa 6 on esitettyä IED:n tallentamasta häiriötallenteesta tehtyjen COMTRADE tiedostojen nimeäminen. /6/



Kuva 6. COMTRADE-standardin mukainen häiriötallenteen nimeäminen. /2/

Otsikkotiedosto on valinnainen, vapaamuotoinen ASCII-tyyppinen COMTRADE datan luoja luoma tekstitiedosto, joka tarjoaa lisätietoa ja auttaa käyttäjää paremmin ymmärtämään tallenteen sisältöä. Se voi esimerkiksi sisältää kuvauksen sähköverkosta ja sähköaseman nimen sekä sen osan nimen, joka häiriön koki. Tiedoston luoja voi melko vapaasti määrätä mitä tietoa ja missä järjestyksessä se otsikkotiedostossa esitetään, koska tietokoneohjelmat eivät muokkaa tai tarvitse tietoa mitenkään. Tiedostopääte on .HDR. /6/

Konfiguraatitiedosto on ASCII-tyyppinen tiedosto, josta ihminen tai tietokone voi tulkita data-arvot siihen liittyvästä datatiedostosta. Konfiguraatitiedosto on tarkasti ennaltamäärätyssä, standardoidussa muodossa, jolloin tietokoneohjelmia ei tarvitse erikseen muunnella jokaista konfiguraatiota varten. Tiedosto sisältää esimerkiksi aseman nimen, kanavien määrät ja tyypit, sekä COMTRADE-standardin version vuoden. Tiedosto on jaettu tarkasti määritellyn muotoisiin riveihin, jotka täytyy esittää tietyssä järjestyksessä, muuten tiedosto mitätöityy. Konfiguraatitiedoston tiedot on lisäksi jaoteltu kriittisiin ja ei-kriittisiin tietoihin. Ei-kriittisiksi luokiteltujen tietojen arvot voidaan jättää tyhjiksi, joskin nämäkin rivit täytyy kirjoittaa. Kriittisen datan puuttuminen tekee tiedostosta

käyttökelvottoman. Alla on esimerkki konfiguraatiotiedoston sisällöstä. Toiseksi alimmaisella rivillä kohdassa ASCII <CR/LF> esimerkiksi määritellään onko datatiedosto ASCII- vai binäärityyppinen. Tiedostopäätte on .CFG. /6/

```

Condie,518,1999 <CR/LF>
12,6A,6D <CR/LF>
1,Popular Va-g,,,kV, 0.14462,0.0000000000,0,Đ2048,2047,2000,1,P
<CR/LF>
2,Popular Vc-g,,,kV, 0.14462,0.0000000000,0,Đ2048,2047,2000,1,P
<CR/LF>
3,Popular Vb-g,,,KV, 0.14462,0.0000000000,0,Đ2048,2047,2000,1,P
<CR/LF>
4,Popular Ia,,,A,11.5093049423,0.0000000000,0,Đ2048,2047,1200,5,P
<CR/LF>
5,Popular Ib,,,A,11.5093049423,0.0000000000,0,Đ2048,2047,1200,5,P
<CR/LF>
6,Popular Ic,,,A,11.5093049423,0.0000000000,0,Đ2048,2047,1200,5,P
<CR/LF>
1,Va over,,,0 <CR/LF>
2,Vb over,,,0 <CR/LF>
3,Vc over,,,0 <CR/LF>
4,Ia over,,,0 <CR/LF>
5,Ib over,,,0 <CR/LF>
6,Ic over,,,0 <CR/LF>
60 <CR/LF>
1 <CR/LF>
6000.000,885 <CR/LF>
11/07/1995,17:38:26.663700 <CR/LF>
11/07/1995,17:38:26.687500 <CR/LF>
ASCII <CR/LF>
1

```

Datatiedosto sisältää jokaisen sisääntulokanavan arvon jokaista tallenteen näytettä kohden. Tallennettu tieto voidaan esittää joko nollapohjaisena, jolloin arvot voivat vaihdella negatiivisen ja positiivisen välillä (esimerkiksi -2000 ja 2000), tai voidaan käyttää niin kutsuttua nolla-offsetiä, jolloin arvot ovat vain positiivisia lukuja, joista jokin valitaan kuvaamaan nollaa (esimerkiksi 0 ja 4000, joista 2000 on nolla). Lisäksi tiedosto sisältää näytteen numeron ja aikaleiman. Datatiedostot voidaan esittää joko binääri- tai ASCII-formaatissa ja kaikki data tiedoston sisällä esitetään kokonaislukuina. Konfiguraatiotiedoston kentässä määritellään kumpaa formaattia käytetään. Alla on esimerkki sekä ASCII-, että binäärityyppisestä datatiedostosta. Tiedostopäätte on .DAT. /6/

```

1, 0, 0994, 1205, 100, 29, 0135, 0197,0,0,0,0,0,0 <CR/LF>
2, 167, 0943, 1231, 94, 37, 0137, 0275,0,0,0,0,0,0 <CR/LF>
3, 333, 0886, 1251, 87, 45, 0139, 0351,0,0,0,0,0,1 <CR/LF>
4, 500, 0826, 1265, 80, 52, 0140, 0426,0,0,0,0,1,0 <CR/LF>
5, 667, 0760, 1274, 72, 61, 0140, 0502,0,0,0,0,1,1 <CR/LF>
6, 833, 0689, 1279, 64, 68, 0140, 0577,0,0,0,0,0,0 <CR/LF>
7, 1000, 0613, 1279, 56, 76, 0139, 0651,0,0,0,0,0,0 <CR/LF>
8, 1167, 0537, 1275, 48, 83, 0139, 0723,0,0,0,0,0,0 <CR/LF>
...
...
883, 147000, 394, 0446, 01, 0, 01, 0345,0,0,0,0,0,0 <CR/LF>
884, 147167, 378, 0417, 02, 0, 01, 0366,0,0,0,0,0,0 <CR/LF>
885, 147333, 360, 0387, 02, 0, 01, 0385,0,0,0,0,0,0 <CR/LF>
<1A><CR/LF>

```

```

01 00 00 00 00 00 00 00 1E FC B5 04 64 00 1D 00 79 FF 3B FF
00 00 02 00 00 00 A7 00 00 00 51 FC CF 04 5E 00 25 00 77 FF
ED FE 00 00 03 00 00 00 4E 01 00 00 8A FC E3 04 57 00 2D 00
75 FF A1 FE 20 00 04 00 00 00 F5 01 00 00 C6 FC F1 04 50 00
34 00 74 FF 56 FE 10 00 05 00 00 00 9C 02 00 00 08 FD FA 04
48 00 3D 00 74 FF 0A FE 30 00 06 00 00 00 43 03 00 00 4F FD
FF 04 40 00 44 00 74 FF BF FD 00 00 07 00 00 00 EA 03 00 00
9B FD FF 04 38 00 4C 00 75 FF 75 FD 00 00 08 00 00 00 91 04
00 00 E7 FD FB 04 30 00 53 00 75 FF 2D FD 00 00 ...

```

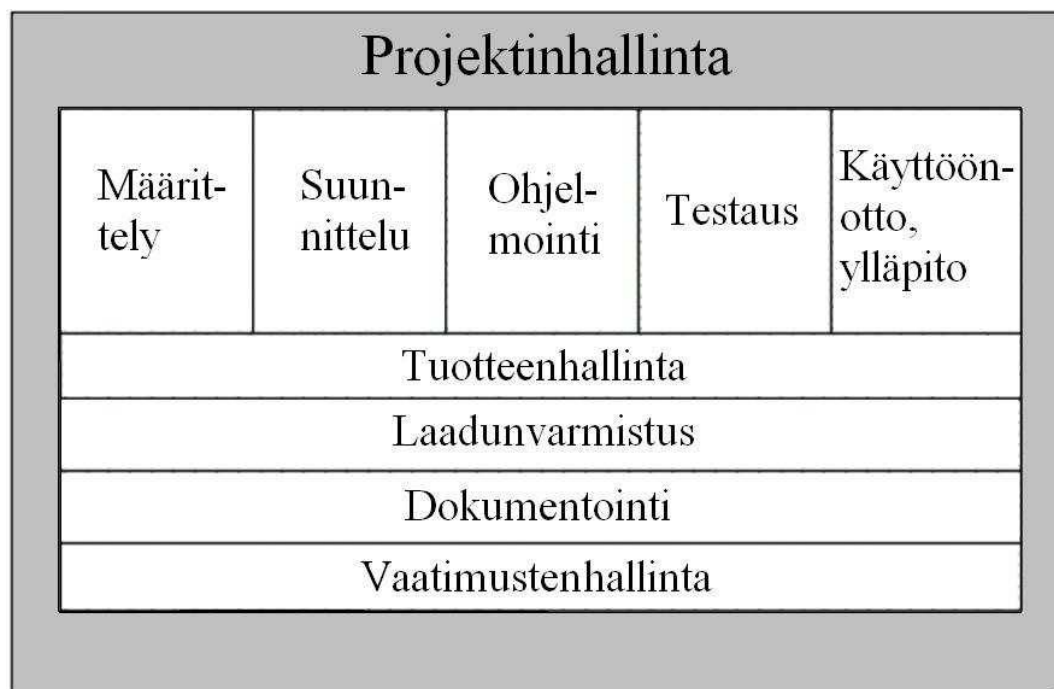
Informaatitiedosto on otsikkotiedoston tavoin valinnainen ASCII-tyyppinen tiedosto, joka sisältää ylimääräistä tietoa, jonka tiedoston luoja saattaa haluta asettaa käyttäjän luettavaksi. Otsikkotiedostosta poiketen informaatiotiedoston rakenne on kuitenkin tarkasti määritelty, jotta tietokoneohjelmien on helpompi lukea sen sisältöä. Formaatti luokittelee tiedolle julkisen osuuden, jota kuka tahansa pääsee lukemaan, sekä yksityisen osuuden, jota vain tietyt käyttäjät pääsevät lukemaan. Informaatitiedosto on jaettu sektioihin, jotka sisältävät uniikin otsikkorivin ja merkintärivejä. Osa informaatiotiedoston tiedoista ovat samoja kuin konfiguraatiotiedostossa. Alla on esimerkki informaatiotiedostosta, joka sisältää julkista ja yksityistä tietoa. Tiedostopäätte on .INF. /6/


```
[Public Record_Information ] <CR/LF>
Source=COMwriter, v1.0<CR/LF>
Record_Information=Fault, AG, Trip, Transmission Line<CR/LF>
Location=189.2, miles<CR/LF>
max_current=3405.5<CR/LF>
min_current=Ð3087.2<CR/LF>
max_voltage=208.6<CR/LF>
min_voltage=Ð206.4<CR/LF>
EventNoteCount=2<CR/LF>
<CR/LF>
...
[Company1 event_rec] <CR/LF>
recorder_type=1<CR/LF>
trig_set=0,0,0,0,6048,6272,0,0,0,0,0,0,0,0,0,0,0,0,0<CR/LF>
ch_type=1,1,1,1,1,1,1,0,0<CR/LF>
<CR/LF>
```

5 TYÖKALU

5.1 Ohjelmistotuotannon vaiheet

Ohjelmistoprojektit saattavat olla todella laajoja, monimutkaisia kokonaisuuksia ja kestää mahdollisesti jopa useita vuosia. Tällaisten projektien kohdalla aseteltujen aikarajojen sekä kustannusten ylittymisten, tai jopa koko projektin epäonnistumisen riskit kasvavat. Tällöin projektinhallinta nousee tärkeään asemaan projektin onnistumisen kannalta. Projektinhallinta tarkoittaa sitä, että projekti jaetaan erillisiin osa-alueisiin, jotka saattavat vaihdella eri projektien välillä, mutta jotka yleensä pääpiirteittäin sisältävät kuvassa 7 esitellyt projektin eri vaiheet.



Kuva 7. Ohjelmistotuotannon osa-alueet. /5/

Ohjelmistotuotannon eri osa-alueet voidaan jakaa kahteen osaan. Kuvassa 2 esitetyssä mallissa niin kutsuttuun kehitysprosessiin kuuluvat määrittely, suunnittelu, ohjelmointi sekä testaus, joita seuraavat myöhemmin tuotteen tai tuoteversion valmistuttua käyttöönotto ja ylläpito. Varsinaiseen toiseen osaan, eli tukitoimintoihin kuuluvat yleensä tuotteenhallinta, laadunvarmistus sekä

dokumentointi. Suuremmissa projekteissa saatetaan erillisinä tukitoimintoina lisäksi pitää esimerkiksi vaatimustenhallintaa ja riskienhallintaa.

Ohjelmistotuotantoa ohjaa yleensä yrityksen oma laatujärjestelmä, joka määrittelee kyseisen yrityksen toimintatavat. Laatujärjestelmän tarkoituksena on yrittää taata, että ohjelmistotuotantoprosessin lopputuloksena syntyvä tuote vastaa suunniteltua laatutasoa ja valmistuu aikataulun ja budjetin mukaisesti. Yrityksen laatuksikirja ja siihen liittyvät dokumentit yleensä kuvaavat yrityksen laatujärjestelmän. Ohjelmistotuotannossakin käytetään laatujärjestelmien arviointiin eri malleja, joista tärkein on ISO 9001 -standardi. Se määrittelee perusasiat, jotka laatujärjestelmän tulee sisältää riippumatta yrityksen toimialasta. /5/

5.2 Määrittely

Ensimmäistä ohjelmistoprojektin vaihetta kutsutaan määrittelyvaiheeksi. Tässä vaiheessa tarkoituksena on selvittää projektin tarpeellisuus ja toteuttamiskelpoisuus, asettaa vaatimukset ja tavoitteet sekä laatia ratkaisumalli. Nämä tulevat yleensä työn tilanneelta asiakkaalta, mutta voivat olla myös esimerkiksi oman suunnitteluryhmän tekemiä.

Vaikka määrittelyvaiheella ei sinänsä ole mitään absoluuttista runkoa, jaetaan se yleensä vaatimusmäärittelyyn, toiminnalliseen määrittelyyn sekä mahdollisesti tekniseen määrittelyyn. Vaatimusmäärittely voi olla esimerkiksi asiakkaan laatima tehtäväkokoelma joihin vaaditaan ratkaisu, tai vaikka tarkka kuvaus valmiin tuotteen edellytyksistä. Toiminnallinen määrittely voidaan jakaa edelleen toiminnallisiin määritelmiin ja ei-toiminnallisiin määritelmiin. Toiminnallinen määrittely tarkoittaa käyttäjän kannalta sovelluksen toimintojen kuvaamista, kuten käyttöliittymää. Ei-toiminnallisiin määritelmiin voidaan laskea esimerkiksi käytettävät rajapinnat, suoritusteho sekä viiveet. Teknisen määrittelyn tarkoituksena yleensä on kuvata toteutusvaiheelle asetettavat vaatimukset. Siinä voidaan kuvata esimerkiksi käytetyt ohjelmointikielet, kehitysympäristöt ja arkkitehtuurit. /5/

Tätä projektia varten laadittiin sisäisesti vaatimusmäärittely ja toiminnallinen määrittely. Lisäksi käytin läpi perusteet tämän projektin aloittamiselle. Vaiheittaisen toteutustavan takia eri vaatimuksille ja toiminnoille annettiin eri tärkeysaste tarpeiden mukaisesti.

5.3 Suunnittelu

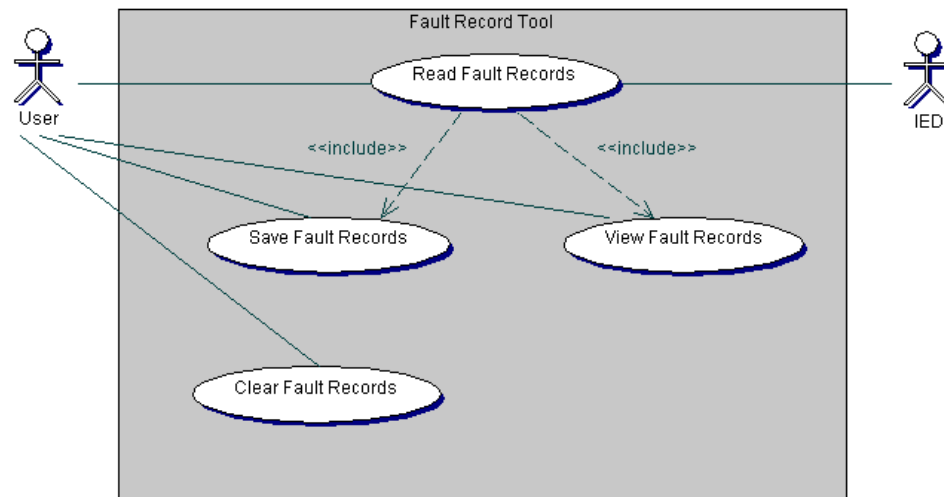
5.3.1 Yleistä

Suunnitteluvaiheessa tehdään aikaisemmassa vaiheessa tehtyjen määrittelyjen perusteella kuvaus järjestelmän toteutuksesta. Suunnittelu jaetaan usein kahteen tasoon: arkkitehtuurisuunnitteluun ja moduulisuunnitteluun. Arkkitehtuurisuunnittelussa määritellään moduulit ja niiden rajapinnat. Tavoitteena on tehdä mahdollisimman vähän toisistaan riippuvia moduuleita, jolloin yksittäisen moduulin sisällä tehtävät muutokset eivät vaikuttaisi tahattomasti muihin moduuleihin. Tämä helpottaa muutosten tekemistä ja ennen kaikkea lisää moduulin uudelleenkäytettävyyttä. Moduulisuunnittelussa keskitytään puolestaan yksittäisiin moduuleihin ja niiden sisäisiin rakenteisiin. /4/

Varsinkin suuremmissa ohjelmistoprojekteissa on tärkeää viedä suunnittelu mahdollisimman pitkälle, koska suunnitteluvaiheessa muutosten tekeminen on huomattavasti helpompaa, halvempaa ja nopeampaa kuin ohjelmointi- ja testausvaiheessa. Tässä tapauksessa kyseessä oli kuitenkin suhteellisen pieni ohjelmistoprojekti, joka ei edellyttänyt monimutkaista projektinhallintaa, joten itse ohjelmointi vei koko projektin kestosta suuremman osan kuin mitä yleisesti ohjelmistoprojektien kohdalla suositellaan.

Tässä opinnäytetyössä keskityttiin suunnittelun osalta lähinnä arkkitehtuurisuunnitteluun. Moduulisuunnittelua esimerkiksi käyttöliittymän kohdalta oli vaikeaa viedä kovin pitkälle, koska ei oltu oikein varmoja miten asiat olisi parasta esittää. Tässä kohtaa päädyttiin tietynlaiseen protoilumalliin, jossa kokeiltiin erilaisia ratkaisuja ja muutoksia tehtiin saadun palautteen perusteella, joten suunnittelua tehtiin osittain ohjelmoinnin yhteydessä. Lisäksi suunnittelua ja ohjelmointia oli tekemässä pääasiassa ainoastaan yksi henkilö. Tämä mahdollisti

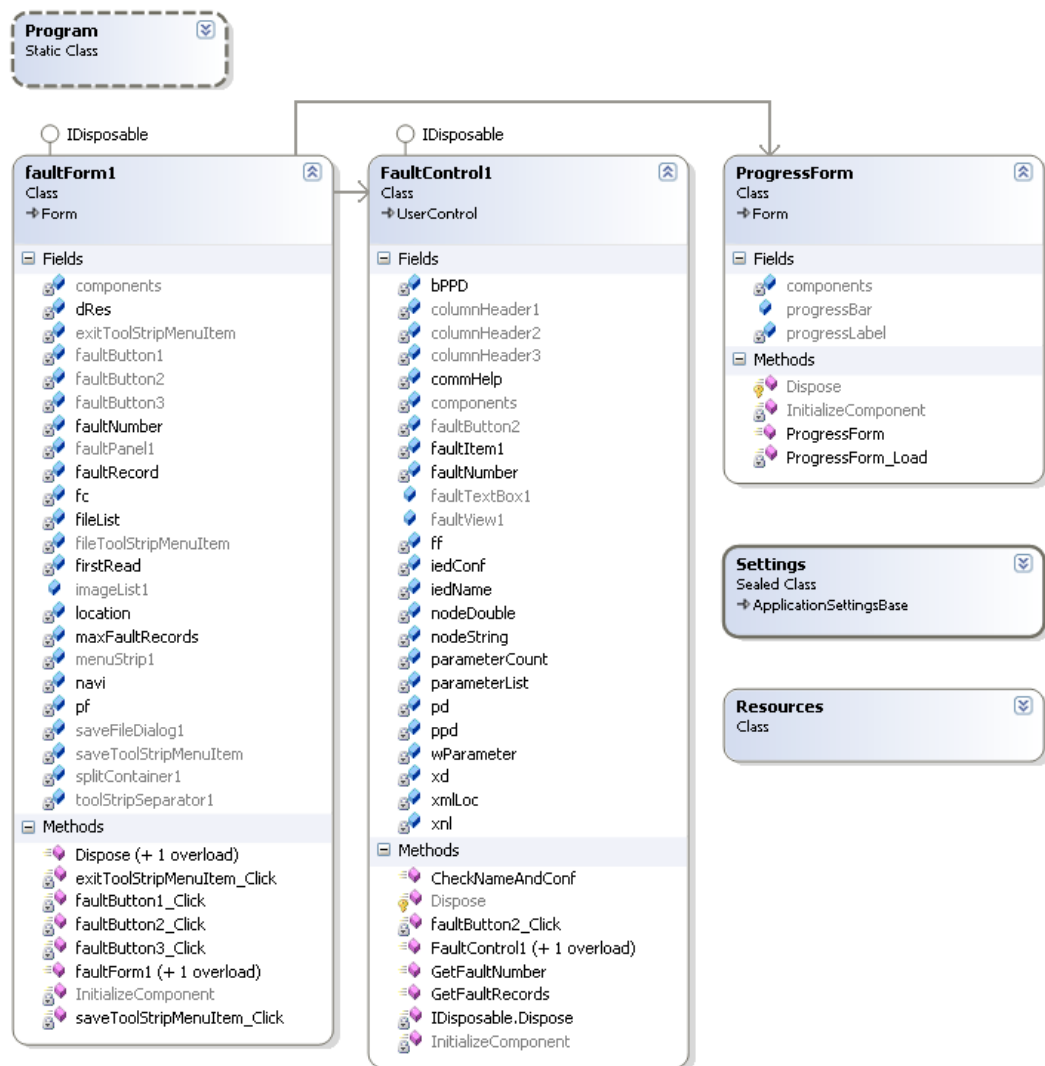
protoiluvaiheen toteuttamisen ilman versionhallintaa, koska ei ollut esimerkiksi sitä riskiä, että eri henkilöiden tekemät muutokset aiheuttaisivat yhteensopivuusongelmia eri moduulien välillä. Näin protoiluvaiheen alusta päästiin nopeammin eteenpäin. Kuvassa 8 on esitettyä työkalun käyttötapakaavio.



Kuva 8. Käyttötapakaavio.

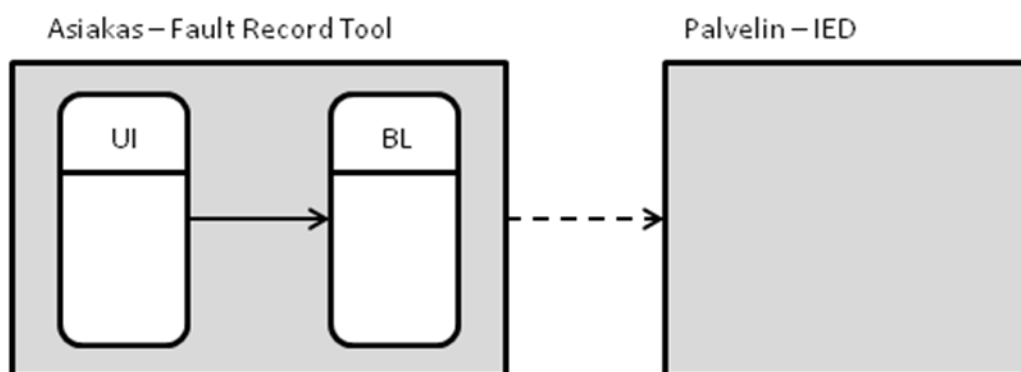
5.3.2 Arkkitehtuurisuunnittelu

Vaikka tämä työkalu on itse asiassa yksittäinen moduuli, joka tulee kuulumaan RE_615-sarjaan, voidaan se kuitenkin tässä tapauksessa ajatella omana kokonaisuutenaan, koska tässä työssä ei oteta sen tarkemmin kantaa koko järjestelmän arkkitehtuuriin ja toimintoihin. Kuvassa 9 on esitettyä sovelluksen luokkakaavio. Sovelluksen toiminnan keskeisimpiä metodeita ovat FaultControl1-luokan lukumetodit.



Kuva 9. Luokkakaavio.

Työkalun ja vikatapahtumatalenteita sisältävän IED:n välinen arkkitehtuuri voidaan ajatella kokonaisuutena asiakas-palvelin -pohjaisena, jossa PC:llä ajettavalla sovelluksella liitytään IED:hen, josta vikatapahtumatalenteet luetaan. Itse sovelluksen arkkitehtuuri on jaettu päätoiminnoiltaan karkeasti kahteen osaan – käyttöliittymään ja liiketoimintaan. Kuvassa 10 on yksinkertainen periaatteellinen kuva arkkitehtuurista.



Kuva 10. Yksinkertainen arkkitehtuurikuva.

5.4 Toteutus

5.4.1 Yleistä

Sovelluksen toteuttamiseen käytettiin Microsoftin kehittämää oliopohjaista C#-ohjelmointikieltä Microsoft .NET-ympäristössä. Yhdistelmä on yleisesti käytössä ohjelmoitaessa Windows-ympäristössä. Sovelluskehityksessä käytettiin Microsoft Visual Studio 2008 -kehitysympäristöä. Itse pääsovelluksen lisäksi tarvittiin vielä Microsoft Office Excel -taulukkolaskentaohjelmassa ajettavaa makroa, joka toteutettiin VBA-ohjelmointikielellä (Visual Basic for Applications) VBE-ympäristössä (Visual Basic Editor). Nämä ovat sisäänrakennettuina Excel -taulukkolaskentaohjelmaan.

Työn edetessä kävi ilmi, että sovelluksen toteutus ja testaus olikin hitaampaa kuin alunperin oli suunniteltu. Tämä tarkoitti sitä, että tähän työhön alunperin suunniteltua toisen vaiheen ominaisuutta ei ehditty toteuttaa, vaan työhön ehdittiin toteuttaa ainoastaan ensimmäisen vaiheen vaatimat toiminnot.

5.4.2 Excel makro

Makrojen tarkoituksena on monissa tapauksissa esimerkiksi automatisoida usein käytettyjä tai itseään toistavia näppäinkomentoja tai hiiren painalluksia, helpottaen ja nopeuttaen kyseisiä toimintoja huomattavasti.

Käytössä olevassa Excel-taulukossa on taulukoituna eri tyyppiset IED:t ja niiden erilaiset konfiguraatiot, sekä kaikkien mahdollisten parametrien nimet, lyhenteet ja yksiköt, sekä merkinnät minkä tyyppiseen IED:hen ja mihin sen konfiguraatioon kukin parametri kuuluu. Näitä parametreja tarvittiin itse pääsovelluksessa, joten tarvittiin keino saada tarvittavat parametrit kerättyä taulukosta sovellukseen. Makron tarkoituksena on huomattavasti helpottaa ja nopeuttaa taulukosta kullekin IED:lle ja sen tietyille konfiguraatiolle kuuluvien parametrien poimintaa, sekä tallentaa ne lisäksi hyvin muodostettuun XML-tiedostoon. Tämä tiedosto luetaan sovellukseen, jossa sen attribuuteista luetaan parametrien nimet, lyhenteet ja yksiköt. Nimet ja yksiköt näkyvät suoraan käyttöliittymässä, mutta lukuoperaatiossa tarvitaan lyhenteitä, joiden avulla saadaan niitä vastaavien parametrien arvot IED:stä sovellukselle. Alla on esimerkki makron luoman XML-tiedoston yhden IED:n yhdestä konfiguraatiosta ja siihen kuuluvista parametreista.


```

<IED type="REF615">
  ...
  <Configuration name="FE06">
    <DO name="StrDur" caption="Start duration" unit="%"/>
    <DO name="ActSG" caption="Setting group" unit=""/>
    <DO name="MaxAmpsA" caption="Max current IL1" unit="xIn"/>
    <DO name="MaxAmpsB" caption="Max current IL2" unit="xIn"/>
    <DO name="MaxAmpsC" caption="Max current IL3" unit="xIn"/>
    <DO name="MaxAmpsN" caption="Max current Io" unit="xIn"/>
    <DO name="VoltsA" caption="Voltage UL1" unit="xUn"/>
    <DO name="VoltsB" caption="Voltage UL2" unit="xUn"/>
    <DO name="VoltsC" caption="Voltage UL3" unit="xUn"/>
    <DO name="VoltsAB" caption="Voltage U12" unit="xUn"/>
    <DO name="VoltsBC" caption="Voltage U23" unit="xUn"/>
    <DO name="VoltsCA" caption="Voltage U31" unit="xUn"/>
    <DO name="VoltsN" caption="Voltage Uo" unit="xUn"/>
    <DO name="AmpsA" caption="Current IL1" unit="xIn"/>
    <DO name="AmpsB" caption="Current IL2" unit="xIn"/>
    <DO name="AmpsC" caption="Current IL3" unit="xIn"/>
    <DO name="AmpsN" caption="Current Io" unit="xIn"/>
    <DO name="AmpsNClc" caption="Current Io-Calc" unit="xIn"/>
    <DO name="AmpsPsSeq" caption="Current Ps-Seq" unit="xIn"/>
    <DO name="AmpsNgSeq" caption="Current Ng-Seq" unit="xIn"/>
    <DO name="VZroSeq" caption="Voltage Zro-Seq" unit="xUn"/>
    <DO name="VPsSeq" caption="Voltage Ps-Seq" unit="xUn"/>
    <DO name="VNgSeq" caption="Voltage Ng-Seq" unit="xUn"/>
    <DO name="DifNAngN" caption="Angle Uo - Io" unit="deg"/>
    <DO name="DifAAngBC" caption="Angle U23 - IL1" unit="deg"/>
    <DO name="DifBAngCA" caption="Angle U31 - IL2" unit="deg"/>
    <DO name="DifCAngAB" caption="Angle U12 - IL3" unit="deg"/>
    <DO name="MaxTmpRl" caption="PTTR thermal level" unit=""/>
    <DO name="PDNS1MxRat" caption="PDNSPTOC1 rat.
I2/I1" unit="%"/>
    <DO name="CondN" caption="Conductance Yo" unit="mS"/>
    <DO name="SusN" caption="Susceptance Yo" unit="mS"/>
  </Configuration>
  ...
</IED>

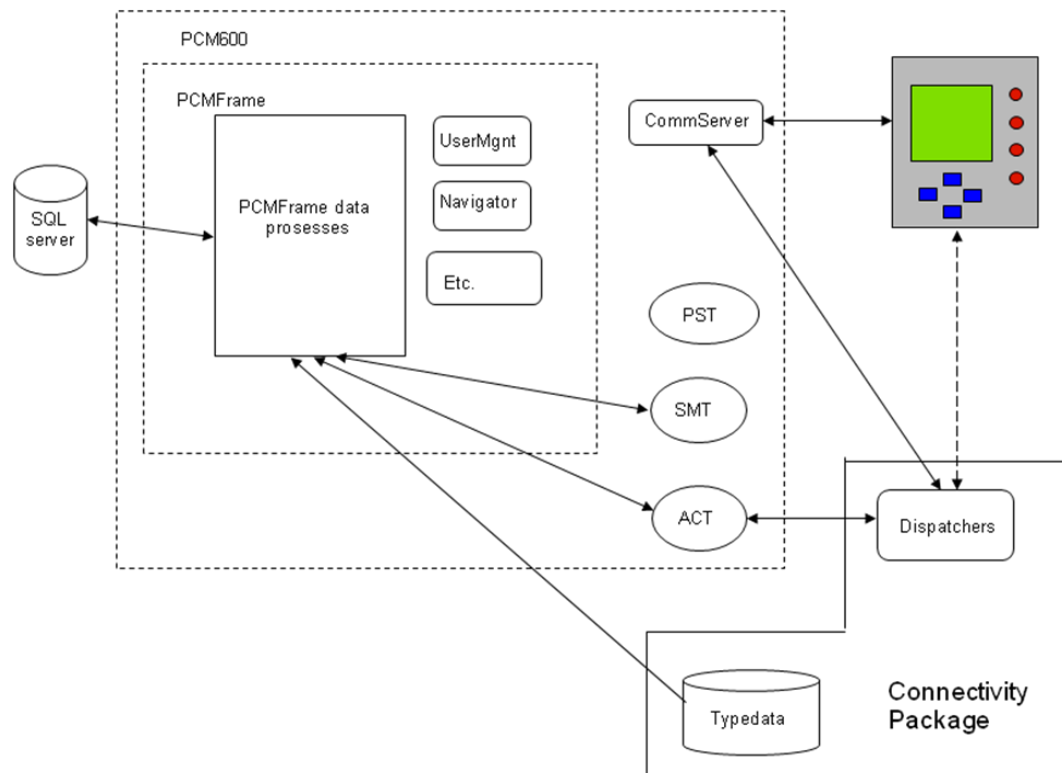
```

Tämä makro ohjelmoitiin jo ennen itse esitystyökalun ohjelmoinnin aloittamista, koska parametreja tultiin tarvitsemaan jo toteutuksen varhaisessa vaiheessa. Täten pystyttiin keskittymään työkalun ohjelmoinnin aloittamisen jälkeen pelkästään sitä koskeviin ongelmiin ja niiden ratkaisuihin.

5.4.3 ConnPack

Sovellus rakennettiin osaksi niin kutsuttuja liitettävyysspaketteja (Connectivity Packages). Ne ovat erikseen asennettavia kokoelmia tiettyyn IED:hen liittyviä sovelluksia ja tietoja, millä tarjotaan muille sovelluksille ja työkaluille mahdollisuus liittyä ja vuorovaikuttaa kyseisen IED:n kanssa. Näitä liitettävyysspaketteja käytetään muutamien muun tuotteen lisäksi PCM600 -

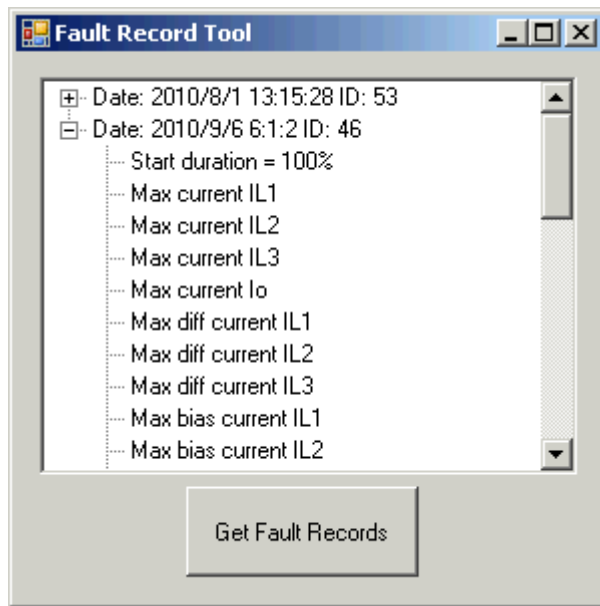
managerointityökalulla, joka sisältää liitettävyyssopakkettien sisältämien sovelluksien lisäksi myös omia työkalukomponentteja. Liitettävyyssopakit sisältävät dataa, joka siirretään PCM600:lle .NET -rajapintojen avulla. Lisäksi ne sisältävät niin kutsuttuja lähettäjiä, jotka ovat vastuussa datan siirtämisestä PCM600:n ja IED:n välillä. Kuvassa 11 on esitetty liitettävyyssopakin ja PCM600:n yksinkertaistettu arkkitehtuuri. /3/



Kuva 11. Liitettävyyssopakin ja PCM600:n yksinkertaistettu arkkitehtuuri. /3/

5.4.4 Fault Record Tool

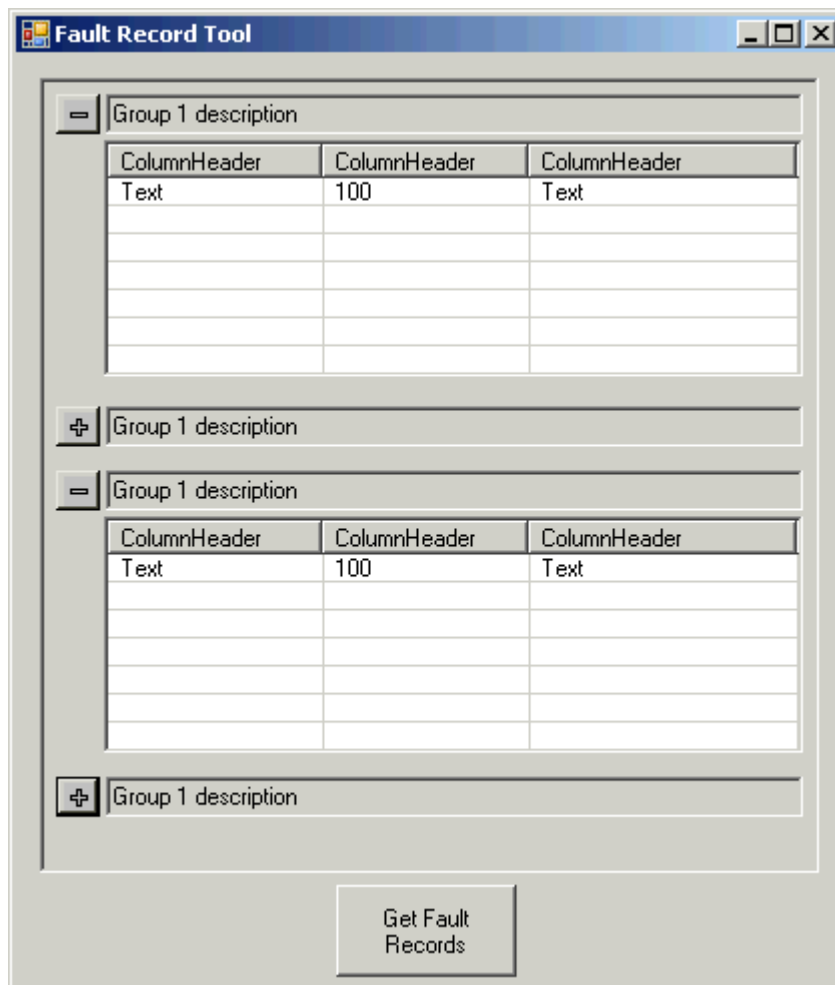
Työn alkuvaiheessa aihetta lähestyttiin puhtaasti käyttöliittymän kannalta, koska haluttiin nopeasti saada ensimmäinen proto tehtyä, jonka avulla saataisiin palautetta siitä, miten itseasiassa vikatapahtumatallenteet olisi viisainta esittää. Ensimmäisessä protossa käytettiin aluksi yksinkertaista TreeView -komponenttia, joka sisälsi juurielementteinä esitettyinä aikaleiman sekä vikatapahtumatallenteen numeron. Näiden lapsielementteinä kuvattiin parametrien nimet, sekä niiden arvot, jotka luettaisiin IED:stä.



Kuva 12. Ensimmäinen proto.

TreeView -komponentin periaate vaikutti hyvältä lähtökohdalta tallenteiden esittämiseen. Eri vikatapahtumatallenteet näkyvät selkeästi alekkain ja niiden sisältämät parametrit saadaan helposti näkyville. Käytyjen keskustelujen jälkeen kävi kuitenkin ilmi, että parametrit ja niiden arvot olisivat selkeämmin luettavissa taulukkomuodossa. Koska yhdestä komponentista rakennettuun puunäkymään ei saa helposti lisättyä taulukkoa, päätettiin TreeView -komponentista luopua kokonaan ja rakentaa käyttöliittymä pienemmistä osista.

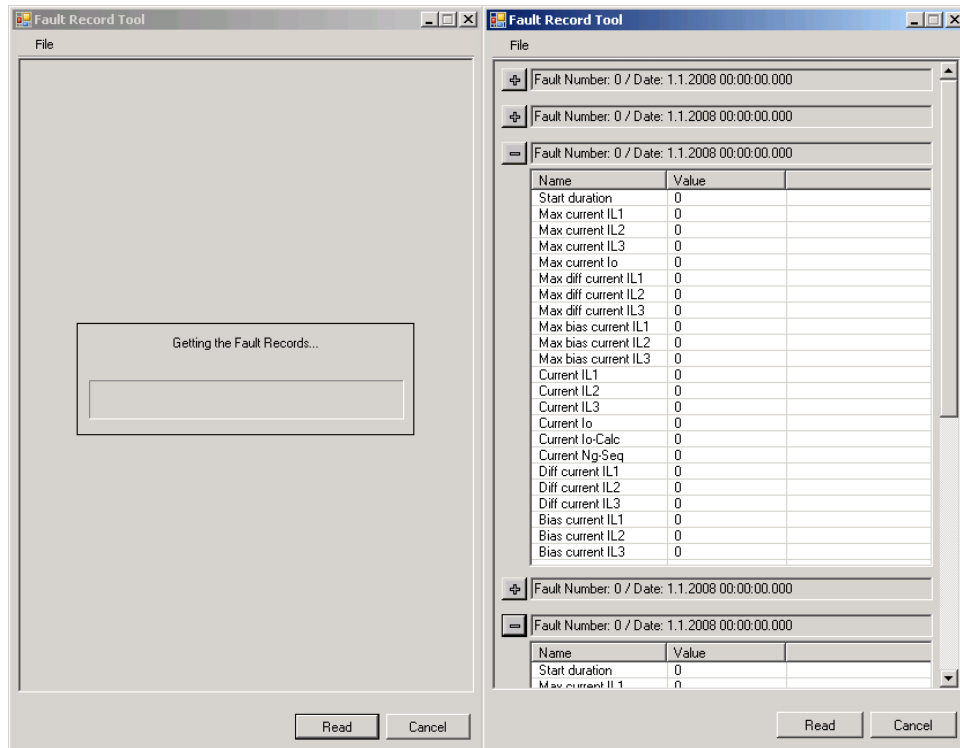
Toiseen protoon otettiin käyttöön erillinen, helposti uudelleenkäytettävä Windows.Forms nimiavaruuteen kuuluva UserControl -käyttöliittymäluokka, joka kuvaisi aina yhtä vikatapahtumatallennetta. Tämä komponentti sisälsi TextBox -komponentin, johon tulisi vikatapahtumatallenteen numero ja aikaleima, ListView -komponentin, johon tulisi taulukoituna parametrien nimet ja arvot, sekä napin, jonka avulla simuloitaisiin puurakenteen laajennus- ja supistustoimintoa muuttamalla UserControlin kokoa. Napin päällä käytettiin ImageList -komponenttia, johon liitettiin kaksi kuvaa kuvaamaan rakenteen eri tiloja.



Kuva 13. Toinen proto.

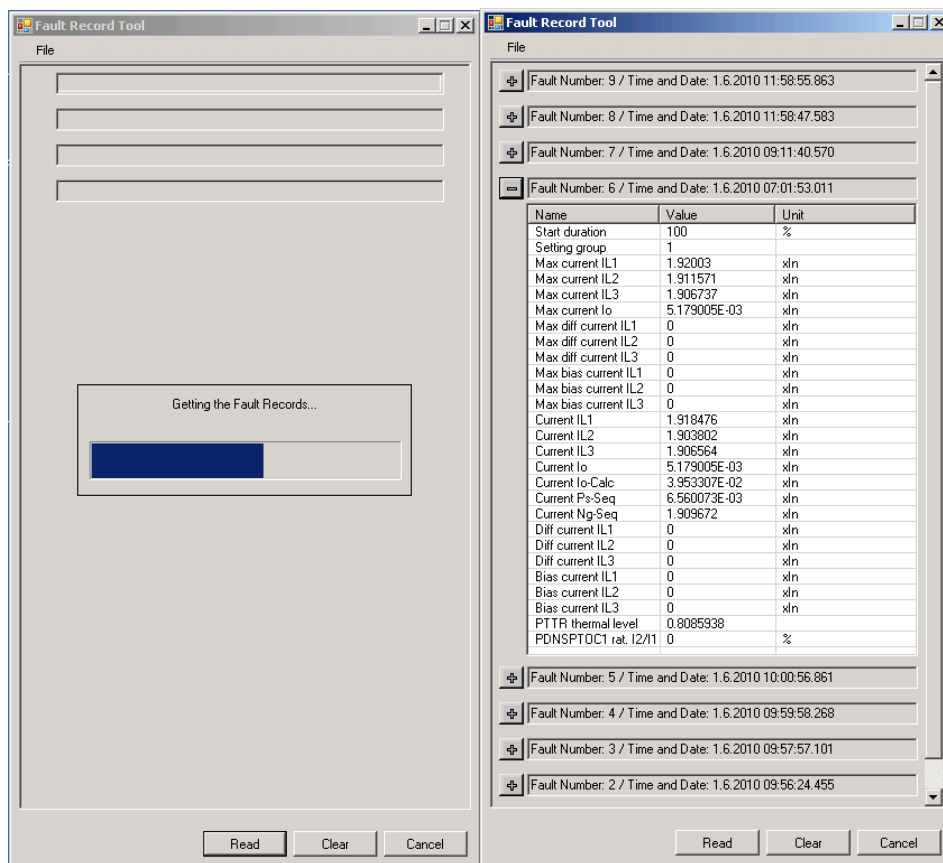
Uusi käyttöliittymä näytti paljon selkeämmältä kuin aikaisempi, joten siihen ei enää suurempia muutoksia toistaiseksi tarvinnut tehdä. Seuraavaksi siirryttiin toiminnallisuuden, eli lukuoperaation ja tallennuksen toteuttamiseen. Lukemiseen tarvittavat metodit toteutettiin FaultControl1 -luokassa. Niissä tarkistetaan vikatapahtumatalenteiden määrä, mikä IED tyyppi ja sen konfiguraatio on kyseessä, sekä täytetään ListView -komponentti tarvittavilla parametreilla. FaultForm1 -luokassa puolestaan luodaan FaultControl1 -luokasta vikatapahtumatalenteiden lukumäärän mukaan tarvittava määrä instansseja, jotka sitten esitetään pääikkunassa. Jotta käyttäjä tietäisi missä vaiheessa lukuoperaatio on menossa, luotiin tätä varten oma ikkuna, jossa esitetään ProgressBar -komponentin avulla lukuoperaation eteneminen. Lisäksi virhetilanteet otetaan

tarvittavissa luokissa kiinni ja niiden virheviestit esitetään varoitusikkunassa. Kuvassa 14 on esitettyä testi IED:n luku ja käyttöliittymään tehdyt muutokset.



Kuva 14. Työkalun toiminnallisuuksia.

Parametrien nimien ja arvojen lisäksi haluttiin vielä lisätä parametrien yksiköt näkyviin. Lisäksi haluttiin lisätä paneelin tyhjäystä varten erikseen nappi. Kuvassa 14 on esitettyä työkalun viimeisin versio, joka tähän opinnäytetyöhön ehdittiin tehdä.



Kuva 15. Työkalun ensimmäisen vaiheen viimeinen versio.

5.5 Testaus

Testausvaiheen tarkoituksena on löytää ohjelmistovirheitä. Prosessimallista riippuen testausvaiheen suunnitteluun voidaan vaikuttaa jo projektin alkuvaiheissa. Tässä opinnäytetyössä testausta suoritettiin kuitenkin jatkuvasti ohjelmoinnin yhteydessä jokaisen lisätyn toiminnon kohdalla, joten erilaisia testitapauksia tai -aineistoja ei erikseen suunniteltu tai käytetty. Ennalta tehtävien täydellisten testitapausten suunnittelu on käytännössä todella hankalaa ja aikaa vievää.

Kun työkalu oli saatu käyttöliittymän osalta miellyttävään muotoon ja ensimmäiset versiot toiminnallisuuksista oli toteutettu, liitettiin työkalu osaksi ConnPackia ja versiohallintaa. Tämä takasi sen, että käyttäjäkokemukset ja työkalun testailu lisääntyivät. Lisäksi automaattinen kääntäjä ilmoitti esimerkiksi yhteensopivuusongelmista muiden työkalujen kanssa. Työkalun testaus ja siihen

tehdyt korjaukset ja parannukset veivätkin enemmän aikaa työstä kuin oli alunperin suunniteltu. Alkuperäisistä suunnitelmista poiketen sovellus jäikin ominaisuuksiltaan suppeammaksi, koska suunniteltua lisätoimintoa ei ehditty annetussa ajassa toteuttaa.

6 YHTEENVETO

Tässä ABB Oy Sähkönjakeluautomaatiolle tehdyssä opinnäytetyössä oli tarkoituksena kehittää vikatapahtumatallenteiden analysointia helpottava työkalu. Työkalun keskeisimmät ominaisuudet ovat vikatapahtumatallenteiden lukeminen IED:stä, niiden esittäminen kootusti selkeässä muodossa graafisessa käyttöliittymässä, sekä tallentaminen käyttäjän tietokoneelle tekstimuotoiseen tiedostoon.

Kirjallisen työn teoreettisessa osuudessa on käyty lyhyesti läpi sähkönjakelun ja relesuojauksen periaatteet, sekä kattavammin vikatapahtumatallenteet ja niihin liittyvä IEEE COMTRADE -standardi. Käytännön osuudessa on käyty läpi ohjelmistoprojektin vaiheet yleisellä tasolla ja tämän projektin kohdalta, sekä esitetty työkalun toimintoja ja käyttöliittymää kuvakaappausten avulla.

Työkalu ohjelmoitiin Microsoftin kehittämällä oliopohjaisella C#-ohjelmointikielellä käyttäen Microsoft .NET-ympäristöä. Työkalu tulee käyttäjän tietokoneelle asennettavien liitettävyysspakettien mukana, joita voidaan käyttää esimerkiksi PCM600 -managerointityökalulla. Työkalun lisäksi tarvittiin Microsoft Office Excel -taulukkolaskentaohjelmassa ajettavaa makroa, joka ohjelmoitiin käyttäen Microsoftin VBA-ohjelmointikieltä.

Lähtökohtana työlle oli alusta asti vaiheittainen toteutustapa. Tarkoituksena oli ensimmäisenä toteuttaa perustoiminnoilla varustettu työkalu, joka tuo apua tärkeimpiin ongelma-kohtiin. Työkalua tullaan jatkokehittämään opinnäytetyön jälkeenkin, joten tämä opinnäytetyö käsitteli työkalun kehitystä ja sen ominaisuuksia niiltä osin kun niitä ajan puitteissa saatiin valmiiksi. Alkuperäisistä suunnitelmista poiketen työkaluun ei ehditty lisätä suunniteltua lisäominaisuutta.

LÄHDELUETTELO

- /1/ ABB Oy 2010. Yleiskatsaus REF615-suojareleeseen [online]. [viitattu 6.4.2010]. Saatavilla www-muodossa: <URL: <http://www.abb.com/product/db0003db004281/c12573e700330419c12573cc00431e30.aspx>>.
- /2/ ABB Oy 615 series, Technical Manual [online]. Päivitetty 2.12.2009 [viitattu 6.4.2010]. Saatavilla www-muodossa: <URL: [http://search.abb.com/LibraryDownloadManager/Default.aspx?resource=http://www05.abb.com/global/scot/scot229.nsf/veritydisplay/efaab01ce4ae6a8fc1257681002bf6f0/\\$File/RE_615_tech_756887_ENb.pdf](http://search.abb.com/LibraryDownloadManager/Default.aspx?resource=http://www05.abb.com/global/scot/scot229.nsf/veritydisplay/efaab01ce4ae6a8fc1257681002bf6f0/$File/RE_615_tech_756887_ENb.pdf)>.
- /3/ ABB Oy PCM600 Simplified Architecture – Yleisesittely PCM600:n arkkitehtuurista ja liitettävyysspaketeista. Julkaisematon esitysmateriaali.
- /4/ ABB Oy Sähköjakeluautomaatio – Yleisesittely. Julkaisematon koulutusmateriaali.
- /5/ Haikala, Ilkka – Märijärvi, Jukka 2000. Ohjelmistotuotanto. 7 p. Pieksämäki. Rt-Print Oy.
- /6/ IEEE Std C37.111-1999, IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems.
- /7/ Intelligent Electronic Device [online]. Päivitetty 16.12.2009 [viitattu 21.4.2009]. Saatavilla www-muodossa: <URL: http://en.wikipedia.org/wiki/Intelligent_electronic_device>.
- /8/ Lakervi, Erkki – Partanen, Jarmo 2008. Sähköjakelutekniikka. Helsinki. Gaudeamus Helsinki University Press.
- /9/ Mörsky, Jorma 1992. Relesuojaustekniikka. 2 p. Espoo. Otatieto.

/10/ Niveri, Henry, suunnitteluarkkitehti, 10.5.2007. ABB Oy
Sähkönjakeluautomaatio, Vaasa. Haastattelu.

LIITELUETTELO

LIITE 1 Ensimmäisen vaiheen vaatimusmäärittely

VAATIMUSMÄÄRITTELY	
Kuvaus	Tärkeys
Vikatapahtumatallenteiden lukeminen IED:stä	1
Käyttöliittymä vikatapahtumatallenteiden tarkastelemiseen tekstimuodossa	1
Vikatapahtumatallenteiden tallentaminen tekstitiedostoon	1