



VAASAN AMMATTIKORKEAKOULU  
VASA YRKESHÖGSKOLA  
UNIVERSITY OF APPLIED SCIENCES

Raine Hermann Laikola

YHTENÄISTETTY VIKATIEDON KE-  
RUU SUOJARELEILTÄ KÄYTÖNTUKI-  
JÄRJESTELMÄÄN

Tekniikka ja liikenne  
2010

## **ALKUSANAT**

Päättötyö on tehty Vaasan ammattikorkeakoulun sähkötekniikan yksikön alaisuudessa ABB:n Sähkönjakelun Automaatiojärjestelmät yksikön tilaamana. Päättötyön aiherajauksesta ja ohjauksesta ovat vastanneet ABB:n taholta projektipäällikkö Esa Markkila ja Vaasan ammattikorkeakoulun taholta sähkötekniikan lehtori Jari Koski. Teknisenä asiantuntijana on toiminut ABB:ltä projektipäällikkö Vesa Lidman. Haluan osoittaa kiitokseni edellä mainituille henkilöille suuresta avusta ja työpanoksesta päättötyön suorittamisessa. Lisäksi haluan kiittää kaikkia mukana olleita henkilöitä, joilta olen saanut arvokkaita neuvoja päättötyön suorittamiseen.

Vaasassa 14.4.2010

Raine Laikola

VAASAN AMMATTIKORKEAKOULU

Sähkötekniikan koulutusohjelma

## TIIVISTELMÄ

Tekijä	Raine Laikola
Opinnäytetyön nimi	Yhtenäistetty vikatiedonkeruu suojarieleiltä käytöntukijärjestelmään
Vuosi	2010
Kieli	suomi
Sivumäärä	59
Ohjaaja	Jari Koski

---

Opinnäytetyössä on tutkittu vianpaikannukseen liittyviä osatekijöitä ja luotu järjestelmien konfigurointia helpottava toiminnallisuus. Työ yhtenäistää vianpaikannukseen liittyvät osatekijät yhdeksi kokonaisuudeksi.

Työssä käytetyt tutkimusmenetelmät pohjautuvat kirjallisen lähdemateriaalin käyttöön ja suoritettuihin haastatteluihin eri yksiköissä. Kirjallinen lähdemateriaali koostuu vianpaikannukseen liittyvästä kirjallisuudesta ja ABB:n eri järjestelmien sekä laitteiden manuaaleista. Haastattelut on suoritettu eri yksiköissä vianpaikannuksen parissa työskentelevien työntekijöiden kanssa. Luotu konfigurointityökalu sisältää valintoja, joilla voidaan vaikuttaa käytettäviin komentoproseduureihin.

Päätötyön tuloksena on yhtenäinen dokumentti, johon on koottu kaikki ABB:n järjestelmissä vianpaikannukseen vaikuttavat osatekijät käytettäessä uutta OPC pohjaista tiedonsiirtoa. Luodun Visual SCIL dialogin avulla järjestelmän konfigurointi helpottuu, sekä komentoproseduurien käyttö on hallitumpaa.

---

Asiasanat                      MicroSCADA, DMS 600, vianpaikannus

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
Sähkötekniikan koulutusohjelma

## ABSTRACT

Author	Raine Laikola
Title	Integrated Collection of Fault Information from Protection Relays to Distribution Network Management System
Year	2010
Language	Finnish
Pages	59
Supervisor	Jari Koski

---

This thesis researches all components which affect fault location process in the medium voltage distribution network. The thesis also includes a configuration tool, which was made during the research. The tool facilitates the configuration of a new systems configuration. The thesis standardizes the fault location process in ABB`s systems.

The research methods used were based on written publications and interviews. The written publications were general publications concerning fault location and ABB`s own system and device manuals. Interviews were had with employees who are working in different units involved in fault location process. The configuration tool created contains selections which have an effect on command procedures used in fault location.

As a result of the this thesis there is a document which collects all fault location information in ABB`s systems, when OPC data access interface is used for data transfer. The Visual SCIL dialog created facilitates the system configuration and makes the use of command procedures more controlled.

---

Keywords                      MicroSCADA, DMS 600, Fault Location

**KÄYTETYT LYHENTEET**

DMS 600	ABB:n käytöntukijärjestelmä
MicroSCADA	ABB:n käytönvalvontajärjestelmä
SYS 600	MicroSCADA perusjärjestelmän tuotetunnus
FLOC	Fault location toimilohko RED 500 sarjan releissä
SCEFRFLO	REF630 fault location toimilohko
OPC	OLE for Process Control, avoimen tiedonsiirron standardi
IEC	International Electrotechnical Commission, kansainvälinen standardointiorganisaatio
RTU	Remote terminal unit, ala-asema tyyppi
Proseduuri	MicroSCADAssa käytettävä komentoketju, yleensä SCIL ohjelmoitu
BDU_FAULT_	Vikatiedon siirrossa käytettävä proseduuri
SCIL	ABB:n MicroSCADA:n käyttämä ohjelmointikieli
Visual SCIL	ABB:n MicroSCADA:n käyttämä ohjelmointikieli visuaaliseen tarkoitukseen
Scil-API	Tiedonsiirtotapa muuttujajoukkotyylisessä vikatiedonsiirrossa
SPA	ABB:n kehittämä liikennöinti-protokolla
GI	General Interrogation, yleiskysely
SAB 600	Station Automation Builder, konfigurointityökalu

## SISÄLLYS

ALKUSANAT.....	2
TIIVISTELMÄ.....	3
ABSTRACT .....	4
KÄYTETYT LYHENTEET .....	5
1 JOHDANTO .....	8
1.1 Opinnäytetyön kuvaus.....	8
1.2 Opinnäytetyön tavoitteet .....	8
1.3 Tutkimusmenetelmät.....	9
2 YLEISTÄ VIANPAIKANNUKSESTA.....	10
2.1 Vianpaikannuksen taustaa .....	10
2.2 Mitattavat viat ja suuret vianpaikannuksessa.....	11
2.2.1 Verkossa ilmenevät viat.....	11
2.2.2 Oikosulku .....	11
2.2.3 Maasulku.....	13
2.3 Vikasuureiden käyttö suojauksessa ja vianpaikannuksessa .....	16
3 VIANPAIKANNUS SUOJARELETASOLLA.....	18
3.1 Yleistä suojaareletason vianpaikannuksesta.....	18
3.2 Vikatiedot REF 543 kennoterminaalissa.....	18
3.2.1 REF 543 Fault location toimilohko .....	19
3.3 Vikatiedot REX 521 suojaareleessä.....	25
3.4 Vikatiedot REF6xx suojaareleessä .....	26
3.4.1 REF630 Fault location toimilohko .....	27
4 VIANPAIKANNUS ALA-ASEMATASOLLA.....	30
4.1 Yleistä ala-asematason kommunikaatiosta.....	30
4.2 Vikapaketin signaalien ristiviittaus Kuumic KU-2000 ala-aseamalla.....	31
4.3 Signaalien ristiviittaus COM 600 ala-aseamalla .....	33
5 VIKATIEDOT KÄYTÖNVALVONTAJÄRJESTELMÄSSÄ.....	38
5.1 Yleistä vikatietojen keruusta käytönvalvontajärjestelmään .....	38
5.2 Komentoproseduurit.....	39
5.4 Luotujen proseduurin kuvaus .....	42

5.5	Vianpaikannuspisteiden ja BDU_FAULT-proseduurin aktivointi .....	45
5.6	Onnistuneiden AJK ja PJK vikatietojen keräys .....	48
5.6.1	Käytännön toteutus onnistuneille PJK ja AJK tiedoille.....	48
5.7	BDU_FAULT proseduurin OPC rajapinnan käyttö.....	49
5.8	Huomioitavaa proseduurien käytössä .....	50
6	VIANPAIKANNUSDIALOGI .....	52
6.1	Visual SCIL .....	52
6.2	Dialogin kuvaus .....	52
7	TULEVAISUUDEN HAASTEET .....	56
8	JOHTOPÄÄTÖKSET .....	57
	LÄHTEET .....	58

# 1 JOHDANTO

## 1.1 Opinnäytetyön kuvaus

Opinnäytetyö tehdään ABB Oy Sähkönjakelun automaatiojärjestelmät yksikölle. Kaikki tutkittavat järjestelmät ovat ABB:n tekemiä. Selkeän dokumentoinnin puutte vianpaikannukseen liittyvistä osatekijöistä muodosti tarpeen päättötyölle. Myös vikatiетоjen siirtoon käytönvalvontajärjestelmästä käytöntukijärjestelmään tarvittavan vikapaketin luontiin ei ollut selkeää ohjetta ja toiminnallisuutta, mikä hidasti järjestelmien konfigurointia. Vianpaikannukseen liittyvät toiminnot koskettavat kaikkia verkostoautomaatiojärjestelmiä toteuttavia osapuolia; rele-, ala-asema-, käytönvalvonta- sekä käytöntukijärjestelmiä. Näiden eri toimitasojen yhtenäistämisen reletasolta valvomoon asti luo helpomman ja varmemman pohjan järjestelmien konfigurointiin. Lisäksi toiminnallisuus, jolla saadaan onnistuneiden pikajälleenkytkentöjen vikatiedot lähetettyä DMS 600 järjestelmään, on hyödyllinen sähkölaitosten verkoissa ilmenevien vikojen ennaltaehkäisyssä.

Työssä kartoitetaan vianpaikannukseen liittyviä ratkaisuja ja ongelmia. Vianpaikannus rajataan keskijänniteverkossa tapahtuviin vikoihin. Työ rajataan järjestelmien osalta vianpaikannukseen relesuojauksen, tiedonsiirtoprotokollien, alasemien, käytönvalvontajärjestelmän sekä käytöntukijärjestelmän osalta. Pääpaino tarkastelussa on mitatun signaalin siirrossa käytöntukijärjestelmälle. Työssä jokainen vianpaikannukseen vaikuttava tekijä on käyty erikseen läpi. Lisäksi luodaan toiminnallisuus, jonka avulla vikapaketin luominen sekä siirto käytöntukijärjestelmälle käytettäessä ABB:n releitä, tapahtuu hallitusti ilman lukuisten erillisten komentoproseduurin luontia.

## 1.2 Opinnäytetyön tavoitteet

Opinnäytetyön tavoitteena on luoda dokumentti, johon on koottuna vikatiетоjen siirtoon suojareleeltä käytönvalvontajärjestelmään vaikuttavat tekijät ja luoda helppokäyttöinen toiminnallisuus, jonka avulla vikatiedot saadaan siirrettyä hallitusti käytönvalvontajärjestelmältä käytöntukijärjestelmälle pakettina.

### **1.3 Tutkimusmenetelmät**

Työssä käytetään lähdemateriaalina vikailmiöihin ja vianpaikannukseen liittyvää kirjallisuutta sekä opetusmateriaaleja. Aikaisemmin tehtyihin ratkaisuihin perehdytään haastattelemalla osaston työntekijöitä sekä etäyhteyden avulla tutustumalla Koillis-Satakunnan Sähkö Oy:n valvomojärjestelmän ratkaisuihin. Lisäksi kartoitetaan DMS 600 järjestelmän tarpeita sekä mahdollisia kehitysideoita vierailemalla käytöntukijärjestelmää kehittävässä ja projektoivassa yksikössä. Työtä tehtäessä ja dokumentoidessa on pyritty keskittymään vianpaikannuksen kannalta oleellisiin seikkoihin ja ratkaisuihin sekä jättämään ylimääräinen perustietojen dokumentointi järjestelmistä vähemmälle osalle.

## 2 YLEISTÄ VIANPAIKANNUKSESTA

### 2.1 Vianpaikannuksen taustaa

Tehokas vianpaikannus on osa turvallista ja laadukasta sähkönjakelua. Sähkön laatuun liittyvät tavoitteet luovat paineita mahdollisimman keskeyttömään sähkönjakeluun. Sähkøyhtiöiden keinot vikojen aiheuttajien eliminointiin ovat rajalliset, mutta erilaisten vikojen nopea paikantaminen pienentää huomattavasti sähkökatkosten kestoa.

Sähkömarkkinalain kolmannen luvun 13 §:n 2 momentin mukaan sähkölaitosten tulee suorittaa raportointi energiamarkkinavirastolle ministeriön määräysten mukaisesti verkkopalvelujensa hintatasoa sekä verkkotoiminnan tehokkuutta, laatua ja kannattavuutta kuvaavista tunnusluvuista. Edellä mainittuun raportointiin liittyy tunnuslukuja, kuten SAIDI, SAIFI ja CAIDI, joilla kuvataan keskeytysten keskimääräistä lukumäärää, yhteenlaskettua kestoaikaa ja keskipituutta. Vikatietojen hallitulla tuomisella käytöntukijärjestelmään helpotetaan myös edellä mainitun raportoinnin suoritusta. /12/

Numeeristen releiden tulo osaksi suojaustoimintoja on mahdollistanut mittaustietojen tehokkaan keräämisen ja prosessoinnin laskennallisesti. Jakeluverkossa syntyneen vian paikallistaminen numeerisen suojarleen mittaustietojen perusteella on lähes poikkeuksetta huomattavasti nopeampaa kuin esimerkiksi vian paikallistaminen kytkinlaitteiden välille kauko-ohjattavien erottimien avulla.

Yleisesti vianpaikannus toteutetaan vertaamalla ennalta verkostolaskennalla saatuja vika-arvoja suojarleeltä saatuihin vikatietoihin. Verrattavat arvot ovat yleisimmin impedanssi tai reaktanssi. Joissain järjestelmissä vianpaikannus perustuu pelkästään vikavirran suuruuteen, tai kaikkiin edellä mainittuihin. /11/

## 2.2 Mitattavat viat ja suureet vianpaikannuksessa

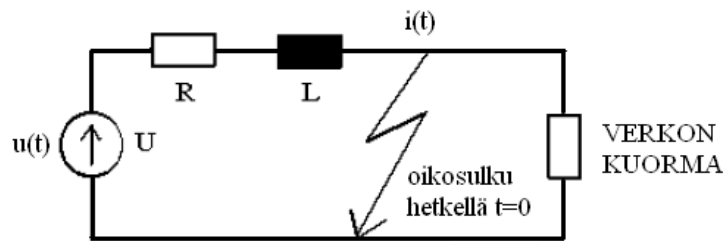
### 2.2.1 Verkossa ilmenevät viat

Käytännössä verkossa ilmenevät viat ovat maasulkuja ja oikosulkuja. Viat voidaan jakaa ohimeneviin ja pysyviin.

### 2.2.2 Oikosulku

Oikosulut aiheutuvat usein vaiheiden välisestä kosketuksesta tai eristeiden epäkunnosta. Oikosulussa tyypillisesti virta ylittää nimellisvirran moninkertaisesti. Oikosulussa esiintyvä virta voidaan jakaa sysäyoikosulkuvirtaan ja jatkuvan tilan oikosulkuvirtaan. Sysäyoikosulkuvirta on huomattavasti suurempi kuin pysyvän tilan oikosulkuvirta.

Oikosulkuvirran teoreettisen tarkastelun avulla voidaan havainnoida eri suureiden muodostumista oikosulussa. Ensimmäisenä muodostetaan yksivaiheinen oikosulkupiiriin sijaiskytkentä, joka on esitettyinä kuvassa yksi.



**Kuva 1.** Oikosulkupiiriin yksivaiheinen sijaiskytkentä.

Syöttöjännitteen hetkellinen arvo saadaan lausekkeesta

$$u(t) = \sqrt{2} * U * \sin(\omega * t + \alpha), \quad (1)$$

jossa  $U$  on vaihejännitteen tehollisarvo,  $\alpha$  on jännitteen vaihekulma ja  $\omega$  on kulmataajuus.

Kirchhoffin jännitelain mukaan saadaan kuvan 1 muutosilmiölle kirjoitettua seuraava differentiaaliyhtälö

$$R * i(t) + L * \frac{di}{dt} = \sqrt{2} * U * \sin(\omega * t + \alpha), \quad (2)$$

jossa R on syöttöpisteen ja oikosulkukohdan välinen reaktanssi, L on syöttöpisteen ja oikosulkukohdan välinen induktanssi, t on kulunut aika oikosulun syntymisestä ja i(t) oikosulkuvirran arvo ajanhetkellä t.

Yhtälön 2 pohjalta ratkaistaan oikosulkuvirran arvo ajanhetkellä t, josta saadaan lauseke

$$i(t) = \sqrt{2} * I_k (\sin(\omega t + \alpha - \phi) - e^{-\frac{t}{\tau}} \sin(\alpha - \phi)), \quad (3)$$

jossa  $I_k$  on pysyvän oikosulkuvirran tehollisarvo, i(t) on oikosulkuvirran arvo,  $\Phi$  on syöttöjännitteen ja oikosulkuvirran vaihe-ero,  $\tau$  on oikosulkuvirran tasakomponentin aikavakio.

Pysyvän oikosulkuvirran tehollisarvo  $I_k$  saadaan lausekkeesta

$$I_k = \frac{U}{Z}, \quad (4)$$

johon impedanssin arvo saadaan kaavasta

$$Z = \sqrt{R^2 + (\omega * L)^2}. \quad (5)$$

Vaihekulma saadaan laskettua lausekkeesta

$$\phi = \arctan \frac{\omega * L}{R}. \quad (6)$$

Aikavakio saadaan laskettua lausekkeesta

$$\tau = \frac{L}{R}. \quad (7)$$

Verkossa tapahtuvan vian aikana voi ilmetä vaimenevaa tasakomponenttia. Mikäli oikosulkuhetkellä pätee

$$\alpha - \phi = 0 + n * \pi, \quad (8)$$

ei tasakomponenttia ole. Kyseessä on täysin symmetrinen vaihtovirta.

Oikosulussa esiintyvä suurin hetkellinen arvo saadaan ratkaistua lausekkeesta

$$i_{s\max} = K * \sqrt{2} * I_k, \quad (9)$$

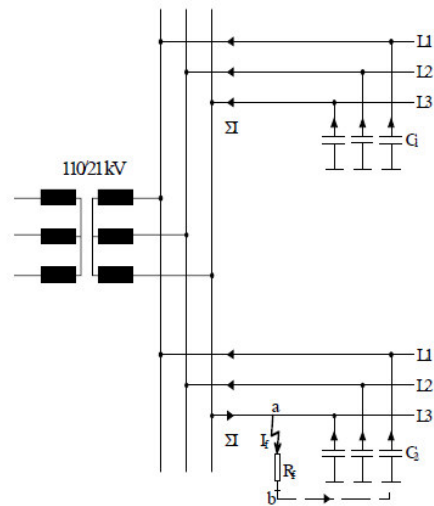
jossa  $I_k$  on pysyvän oikosulkuvirran tehollisarvo ja K on sysäyskerroin, joka riippuu suhteesta R/X.

Relesuojauksen kannalta tärkeä arvo on pysyvän oikosulkuvirran tehollisarvo  $I_k$ . Arvon kasvaessa releellä aseteltuja laukaisu-arvoja suuremmiksi vikatilanteessa, suojaustoiminnot havahtuvat.

### 2.2.3 Maasulku

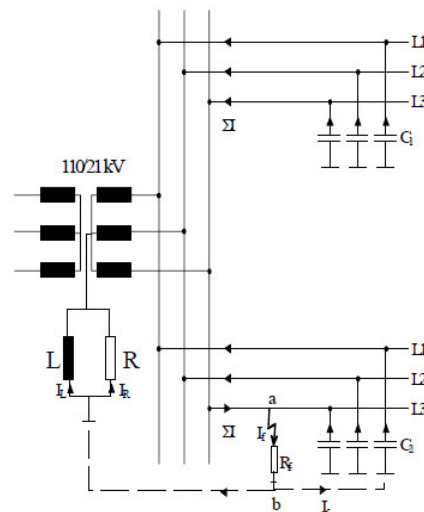
Maasulku tapahtuu vaiheen ja maapotentiaalin väliin syntyvän yhteyden kautta. Keskijänniteverkon vioista maasulkuja on noin 60 – 70 %:a. Maasulkuviat voidaan jakaa neljään eri luokkaan: yksivaiheinen maasulku, kaksoismaasulku, kaksivaiheinen maaosulku sekä johdinkatkeaman aiheuttama yksivaiheinen maasulku kuorman puolella. /16/

Pääsääntöisesti sähköverkot ovat Suomessa maasta erotettuja (kelluva verkko) tai sammutuskuristimen kautta maadoitettuja (sammutettu verkko). Maasta erotetussa verkossa vikavirralla ei löydy pieni-impedanssista kulkureittiä. Tästä syystä vikavirralla on kulkureitti ainoastaan maakapasitanssien kautta. Sivulla 15 olevassa kuvassa 2 on esitettyä maasta erotetun verkon maasulku.



**Kuva 2.** Maasta erotetun verkon maasulku.

Sammutetussa verkossa vikavirta kulkee säädettävän sammutuskuristimen kautta. Tähtipisteeseen maadoitetun reaktanssin avulla kompensoidaan verkossa olevat kapasitiiviset loisvirrat. Lisäksi kuristimen avulla saadaan pienennettyä tai jopa sammutettua maasulkuvirta. Kuvassa 3 on esitettyä sammutetun verkon maasulku.



**Kuva 3.** Sammutetun verkon maasulku.

Laskennallisesti maasulussa syntyvä maasulkuvirta saadaan lausekkeesta

$$I_E = \frac{U_V}{1} = \sqrt{3} * \varpi * C_0 * U . \quad (10)$$

$$\frac{1}{3 * \varpi * C_0}$$

Lausekkeen käyttö edellyttää vikaresistanssin olevan nolla. Mikäli vikaresistanssia esiintyy, käytetään lauseketta

$$I_{EF} = \frac{I_{EM}}{\sqrt{1 + \left( \frac{\sqrt{3} * I_{EM} * R_F}{U} \right)^2}}, \quad (11)$$

jossa  $I_{EM}$  on maasulkuvirta vikaresistanssin ollessa nolla,  $R_F$  on maasulkupiirin vikaresistanssi ja  $U$  nimellinen pääjännite. Kyseinen lauseke ei ota huomioon verkossa olevia nollakapasitansseja. Otettaessa  $C_0$  nollakapasitanssit huomioon, käytetään lauseketta

$$I_{EF} = \frac{U_V}{Z_{CF}} \frac{\frac{U}{\sqrt{3}}}{\sqrt{R_F^2 + \left( \frac{1}{3 * \varpi * C_0} \right)^2}} . \quad (12)$$

Maasulkuvirta syntyy viassa tapahtuvan vaiheiden jännitteiden epäsymmetrian vuoksi. Epäsymmetrian takia verkon tähtipisteeseen muodostuu nollajännite. Nollajännitteellä tarkoitetaan verkon tähtipisteen ja maan välistä jännitettä. Verkon ollessa normaalitilassa, pysyy nollajännite miltei nollana. Vikahetkellä nollapiste lähtee vaeltamaan kohti maapistettä. Tällöin terveiden vaiheiden jännite nousee rajusti. Verkon jäännösvirta ja nollajännite maasta erotetussa verkossa saadaan lausekkeilla

$$I_{mf} = \frac{3 * \varpi * C_0}{\sqrt{1 + (3 * \varpi * C_0 * R_f)^2}} * \frac{U}{\sqrt{3}} \quad (13)$$

$$U_0 = \frac{1}{\sqrt{1 + (3 * \varpi * C_0 * R_f)^2}} * \frac{U}{\sqrt{3}} . \quad (14)$$

Verkon jäännösvirta ja nollajännite sammutetussa verkossa saadaan lausekkeilla

$$I_{mf} = \frac{\sqrt{1 + R_0^2 * (3 * \omega * C_0 - \frac{1}{\omega L})^2}}{\sqrt{(R_f + R_0)^2 + R_f^2 * R_0^2 * (3 * \omega * C_0 - \frac{1}{\omega * L})^2}} * \frac{U}{\sqrt{3}}, \quad (15)$$

$$U_0 = \frac{1}{\sqrt{(1/R_0)^2 + (3 * \omega * C_0 - \frac{1}{\omega * L})^2}} * I_{mf}, \quad (16)$$

jossa  $R_0$  on kompensointikuristimen ja verkon häviöitä vastaavan resistanssin sekä mahdollisen kuristimen toisioresistanssin tähtipisteeseen redusoitu kokonaisresistanssi. Nollavirta saadaan kaavasta

$$I_0 = \frac{3 * C_0 - 3 * C_{0j}}{3 * C_0} * I_{mf}, \quad (17)$$

jossa  $C_0$  on verkon yhden vaiheen maakapasitanssi ja  $C_{0j}$  on suojattavan johdon yhden vaiheen maakapasitanssi.

### 2.3 Vikasuureiden käyttö suojauksessa ja vianpaikannuksessa

Vianpaikannuksen kannalta tärkeitä suureita ovat virta ja jännitetiedot. Vianpaikannus voidaan toteuttaa pelkästään virtamittauksen perusteella, jolloin suojareleen rekisteriin saatu vikavirtatieto lähetetään ala-aseman ja käytönvalvontajärjestelmän kautta käytöntukijärjestelmään. Saatuja arvoja verrataan käytöntukijärjestelmässä jo ennalta laskettuihin vika-arvoihin. Tämän tyyppinen vianpaikannus ei ota huomioon oikosulussa olevaa vikaresistanssia, mikä aiheuttaa pienen vääristymän vianpaikannukseen. /11/

Yleisimmin käytetty paikannusmenetelmä perustuu jännitteen ja virran mittaukseen, jossa lasketaan vianaikainen reaktanssi. Tässä työssä tutkittava ABB:n DMS 600 pohjainen vianpaikannus perustuu ratkaisuun, jossa paikannukseen käytettävä keino riippuu saadun datan määrästä. Käytöntukijärjestelmän minimiratkaisussa ilmoitetaan vikaantunut lähtö. Tähän minimiratkaisuun tarvitaan lopullisen laukai-

sun tapahtuma-aika, vikaantuneen lähdön katkaisijan looginen nimi, sekä triggauksen tieto, joka ilmaisee laukaisun tyypin ("unknown", "final", "reclosing", "delayed\_reclosing"). Kriittisen datamäärän kasvaessa voidaan ottaa käyttöön vikavirtaan sekä reaktanssiin perustuvia paikannusmenetelmiä, joihin tarvitaan vian tyyppi, vaihetieto, vikavirrat ja reaktanssitieto. Tietyissä tapauksissa täytyy vikapakettiin sisällyttää kuormitusvirrat, koska riittävän suuri kuormitusvirta voi vääristää paikannusta. /1/

### 3 VIANPAIKANNUS SUOJARELETASOLLA

#### 3.1 Yleistä suojarahetason vianpaikannuksesta

Suojareleillä voidaan havaita verkossa tapahtuvia vikoja, jännitteiden ja virtojen vaihteluita sekä valvoa sähkön laatua ja ohjata kytkinlaitteita. Yleisesti suojarahetason toimintoja ovat suojaus, mittaus, sähkön laadunvalvonta, ohjaus, kunnonvalvonta, tietoliikenne, yleistoinnot ja perustoiminnot. Tässä työssä tutkittavien releiden osalta syvennytään mittaus- ja suojauslohkojen toimintaan. Vikatilanteissa suojaraheltä tarvittava tieto on kaikki vianpaikannuslohkolta saatava data, tai kuormitus- ja vikavirrat, riippuen suojarahetason tyyppistä. Edellä mainituista saadaan kerättyä vikapaketti käytöntukijärjestelmän tarpeisiin. Yleensä vikapaketissa siirretään kaikki vikatilanteista mitattu data, vaikkei itse laskentatoimintoihin käytettäisi kuin virtaa tai reaktanssia.

#### 3.2 Vikatiedot REF 543 kennoterminaalissa

REF 543 kennoterminaalissa vikapaikannuksen kannalta tärkeitä toimintoja ovat suojaus- mittaus- sekä vianpaikannustoiminnot. Rele tukee SPA, LON, IEC-103, DNP 3.0 ja Modbus protokollat. Erillisen SPA-ZC muuntimien kautta releeltä tulevaa protokollaa voidaan muuntaa edellä mainittujen lisäksi IEC 61850 protokollaksi. Kuvissa 4 ja 5 olevissa taulukoissa ovat esitettyinä kaikki releessä olevat mittaus- ja suojaustoiminnot. /3/

Toiminto	Selitys
MEA1 <sup>2)</sup>	Mittausulo 1 / analogiatulo RTD-/analogiamoduuli
MEA2 <sup>2)</sup>	Mittausulo 2 / analogiatulo RTD-/analogiamoduuli
MEA3 <sup>2)</sup>	Mittausulo 3 / analogiatulo RTD-/analogiamoduuli
MEA4 <sup>2)</sup>	Mittausulo 4 / analogiatulo RTD-/analogiamoduuli
MEA5 <sup>2)</sup>	Mittausulo 5 / analogiatulo RTD-/analogiamoduuli
MEA6 <sup>2)</sup>	Mittausulo 6 / analogiatulo RTD-/analogiamoduuli
MEA7 <sup>2)</sup>	Mittausulo 7 / analogiatulo RTD-/analogiamoduuli
MEA8 <sup>2)</sup>	Mittausulo 8 / analogiatulo RTD-/analogiamoduuli
MEAO1 <sup>2)</sup>	Analogialähtö 1 RTD-/analogiamoduuli
MEAO2 <sup>2)</sup>	Analogialähtö 2 RTD-/analogiamoduuli
MEAO3 <sup>2)</sup>	Analogialähtö 3 RTD-/analogiamoduuli
MEAO4 <sup>2)</sup>	Analogialähtö 4 RTD-/analogiamoduuli
MECU1A	Nollavirtamittaus A
MECU1B	Nollavirtamittaus B
MECU3A	Kolmivaiheinen virtamittaus A
MECU3B <sup>2)</sup>	Kolmivaiheinen virtamittaus B
MEDREC16 <sup>1)</sup>	Transienttia mittaava häiriötalennin
MEFR1	Järjestelmätaajuuden mittaus
MEPE7	Kolmivaiheinen tehon- ja energiamittaus sekä tehokerroin
MEVO1A	Nollajännitemittaus A
MEVO1B <sup>2)</sup>	Nollajännitemittaus B
MEVO3A	Kolmivaiheinen jännitemittaus A
MEVO3B <sup>2)</sup>	Kolmivaiheinen jännitemittaus B

**Kuva 4.** REF 543 kennoterminaalien mittaus- ja suojaustoiminnot.

Toiminto	Selitys
AR5Func	Jälleenkytkentä (5 porrasta)
CUB1Cap <sup>2)</sup>	Kondensaattoriparistojen epäsymmetriasuoja
CUB3Cap <sup>3)</sup>	H-siitään kytketyn kondensaattoripariston 3-vaiheinen epäsymmetriasuoja
CUB3Low	Vaihekatkosuoja
DEF2Low	Suunnattu maasulkusuoja, I <sub>0</sub> -porras
DEF2High	Suunnattu maasulkusuoja, I <sub>0</sub> >>-porras
DEF2Inst	Suunnattu maasulkusuoja, hetkellistoiminta
DOC6Low <sup>1)</sup>	Kolmivaiheinen, suunnattu ylivirtasuoja, I->-porras
DOC6High <sup>1)</sup>	Kolmivaiheinen, suunnattu ylivirtasuoja, I->>-porras
DOC6Inst <sup>1)</sup>	Kolmivaiheinen, suunnattu ylivirtasuoja, hetkellistoiminta
Freq1St1 <sup>1)</sup>	Alli- tai ylitaajuussuoja, porras 1
Freq1St2 <sup>1)</sup>	Alli- tai ylitaajuussuoja, porras 2
Freq1St3 <sup>1)</sup>	Alli- tai ylitaajuussuoja, porras 3
Freq1St4 <sup>1)</sup>	Alli- tai ylitaajuussuoja, porras 4
Freq1St5 <sup>1)</sup>	Alli- tai ylitaajuussuoja, porras 5
Fusetaj <sup>3)</sup>	Sulakevikavalvonta
Inrush3	Kolmivaiheinen, muuntajan kytkentä- ja moottorin käynnistysvirran ilmaisin
MotStart <sup>2)</sup>	Moottorien kolmivaiheinen käynnistysvirran valvonta
NEF1Low	Suuntaamaton maasulkusuoja, I <sub>0</sub> -porras
NEF1High	Suuntaamaton maasulkusuoja, I <sub>0</sub> >>-porras
NEF1Inst	Suuntaamaton maasulkusuoja, hetkellistoiminta
NOC3Low	Kolmivaiheinen, suuntaamaton ylivirtasuoja, I->-porras
NOC3High	Kolmivaiheinen, suuntaamaton ylivirtasuoja, I->>-porras
NOC3Inst	Kolmivaiheinen, suuntaamaton ylivirtasuoja, hetkellistoiminta
OL3Cap <sup>2)</sup>	Kondensaattoripariston 3-vaiheinen ylikuormitusuoja
OV3Low	Kolmivaiheinen ylijännitesuoja, U->-porras
OV3High	Kolmivaiheinen ylijännitesuoja, U->>-porras
PSV3St1 <sup>2)</sup>	Symmetrisiin komponentteihin perustuva jännitesuoja, porras 1
PSV3St2 <sup>2)</sup>	Symmetrisiin komponentteihin perustuva jännitesuoja, porras 2
ROV1Low	Nollajännitesuoja, U <sub>0</sub> -porras
ROV1High	Nollajännitesuoja, U <sub>0</sub> >>-porras
ROV1Inst	Nollajännitesuoja, hetkellistoiminta
SCVCS1 <sup>1)</sup>	Tahdissaaton valvontatoiminto/jännitevalvontatoiminto, porras 1
SCVCS2 <sup>1)</sup>	Tahdissaaton valvontatoiminto/jännitevalvontatoiminto, porras 2
TOL3Cab <sup>1)</sup>	Kolmivaiheinen terminen kaapelisuoja
TOL3Dev <sup>2)</sup>	Kolmivaiheinen terminen laitesuoja
UV3Low	Kolmivaiheinen alijännitesuoja, U->-porras
UV3High	Kolmivaiheinen alijännitesuoja, U->>-porras

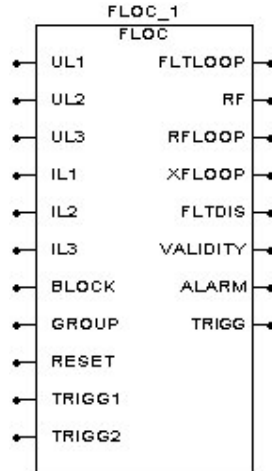
**Kuva 5.** REF 543 kennoterminaalin suojaustoiminnot.

### 3.2.1 REF 543 Fault location toimilohko

FLOC (Fault location) toimilohkon avulla voidaan suorittaa erilaisia vianpaikannukseen tarvittavia laskentoja jo releessä sisäisesti. REF 543 suoja-releessä käytettävä FLOC-lohko on RED 500 sarjan releisiin toteutettu ratkaisu. Toimilohkon avulla voidaan lähettää vianpaikantamiseen tarvittavia tietoja tai lähettää vikaetäisyys suoraan FLTDIS (Fault distance) tietona. Lohko tarvitsee laskentoihin kaikki vaihevirratt ja vaihejännitteet. Vianpaikannus koostuu kahdesta eri vaiheesta, vian tyyppin ratkaisusta ja vikapaikan laskennasta. Vikatyyppi ratkaistaan toimilohkoon sisäänrakennetun Phase Selection Logic (PSL) toiminnolla. Tarkin lopputulos saadaan kun FLOC-lohkon tiedot lähetetään käytöntukijärjestelmään, joka suorittaa tarkemman vertailun ja laskennan. /2/

Releen konfigurointi työkalussa toimilohkon vasempaan reunaan linkitetään mitaustiedot vaihejännitteistä ja vaihevirroista sekä muut lohkon käyttöön vaikutta-

vat signaalit, kuten esto- ja laukaisusignaalit, jotka ovat esitettyinä kuvassa 6. Toimilohkon oikeassa reunassa ovat laskentaprosessien perusteella saatavat ulostulot, jotka ovat esitettyinä kukin erikseen kuvassa 7. /2/



**Kuva 6.** Fault location toimilohko.

Name	Type	Description
FLTLOOP	Short integer (SINT)	Fault impedance loop used for distance measurement.
RF	Analog signal (REAL)	Fault resistance in primary ohms.
RFLOOP	Analog signal (REAL)	Fault loop resistance in primary ohms.
XFLOOP	Analog signal (REAL)	Fault loop reactance in primary ohms.
FLTDIST	Analog signal (REAL)	Fault distance in per unit (pu). Pu is the distance unit selected by the user.
VALIDITY	Short integer (SINT)	Result validity indicator for earth faults.
ALARM	Digital signal (BOOL, pos. edge)	Alarm signal indicating that the fault is located between the set limits.
TRIGG	Digital signal (BOOL, pos. edge)	Signal indicating the triggering of registers and the outputs of the function.

**Kuva 7.** Fault location toimilohkon ulostulot.

Vian tyyppin laskenta suoritetaan vikasilmuksen valintaa varten. Lohkossa on käytössä seitsemän eri vikalaskentasilmuksia, jotka ovat esitettyinä sivulla 22 olevassa kuvassa 8. Käytännön erona laskentasilmuksissa ovat maasulussa laskentaan mukaan otettavat komponentit  $R_N$  ja  $X_N$ , jotka ovat viassa maan kautta kiertävän piirin resistanssi ja reaktanssi. /2/

Silmukoita 1, 2 ja 3 käytetään, mikäli vika on tyypiltään maasulku jollakin vaiheella. Kyseisiä silmukoita käytetään myös, mikäli kyseessä on kahden vaiheen aiheuttama maasulku, jotka ovat eri lähdöillä. /2/

Silmukoita 12, 23 ja 31 käytetään oikosulkutilanteissa sekä kahden vaiheen aiheuttamassa maasulussa, mikäli vika tapahtuu samalla johtolähdöllä. /2/

Fault type	Description	Fault loop
L1-E	Phase L1-to-earth fault	1
L2-E	Phase L2-to-earth fault	2
L3-E	Phase L3-to-earth fault	3
L1-L2	Phase L1-to-L2 short circuit fault	12
L2-L3	Phase L2-to-L3 short circuit fault	23
L3-L1	Phase L3-to-L1 short circuit fault	31
L1-L2-L3-(E)	Three-phase short circuit	123

**Kuva 8.** Fault location toimilohkon vikalaskentasilmukat.

Vianpaikannuksessa tulee huomioida myös tarkkuuteen vaikuttavat tekijät. Vikaresistanssin ollessa pieni, pysyy laskenta tarkkana. Virta- ja jännitemuuntajien kyllästyminen vikailmiön aikana aiheuttavat virhettä laskentaan. Kyllästyminen saattaa myös lisätä tiettyjen harmonisten yliaaltojen esiintymistä. FLOC-lohko sisältää validity-toiminnon, jonka perusteella saadaan tietoa vikapaikannuksen luotettavuudesta. Edellä mainittujen osatekijöiden perusteella validity ulostulo antaa tietoa laskentatarkkuudesta. /2/

Paikannuksen laatu voi olla tasoltaan

- 1) N/A (0). Vian tyyppi on oikosulku, jolloin ei ole tarvetta tarkkuusarvioinnille.
- 2) High (1). Paikannuksen tarkkuuteen ei ole vaikuttanut häiritseviä tekijöitä.
- 3) Moderate (2). Paikannuksen tarkkuus on hieman kärsinyt virheellisten tietojen takia. Virhemarginaali voi olla kymmeniä prosentteja.

- 4) Poor (3). Vianpaikannus algoritmeihin vaikuttaa voimakkaasti virheelinen data, eikä laskentaa voida suorittaa tarkasti. Tällaisessa tapauksessa paikannus on karkeasti suuntaa antava. Paikannus voidaan tehdä lähinnä tutkimalla kummassa päässä johtolähtöä vika on.

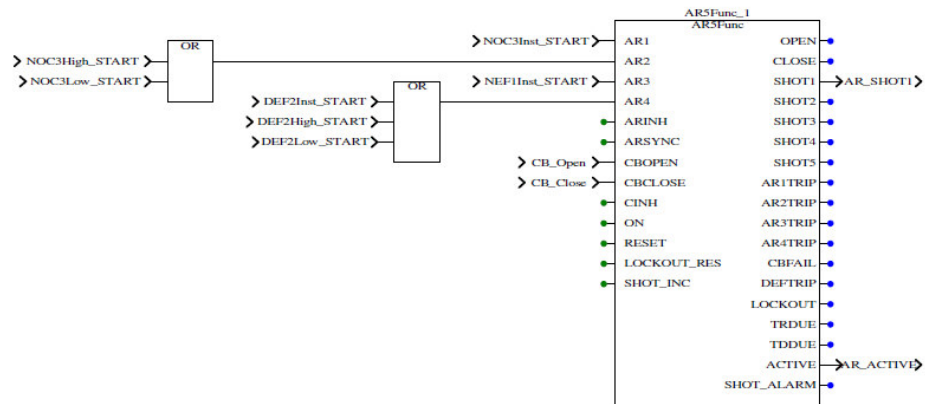
Käytettäessä FLOC- toimilohkoa, voidaan lohkon liipaisu toteuttaa kolmella eri tavalla.

Ensimmäinen vaihtoehto on ulkoinen liipaisu, jossa FLOC-lohkon toimintaa ohjataan ulkoisen signaalin avulla. Tällaisessa ratkaisussa liipaisu tapahtuu jonkin suojauslohkon trip signaalin avulla. Kun tietty suojaus alkaa toimia, toimii myös FLOC-lohko välittömästi. /2/

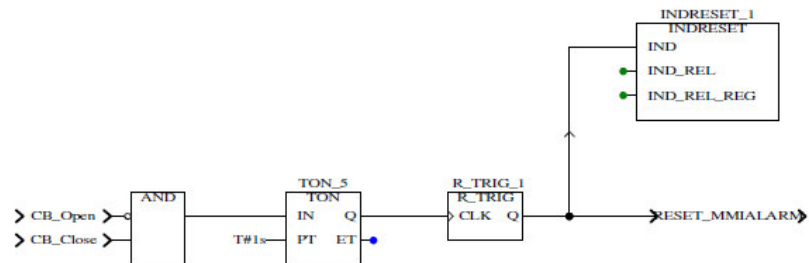
Toinen vaihtoehto on sisäinen liipaisu, jolloin trigg1 tulonastaan ei liitetä mitään. Liipaisu saadaan sisäisesti PSL (Phase selection logic) vikatyypin laskenta toiminnon avulla. Tätä vaihtoehtoa käytettäessä tulee kuitenkin huomioida eri suojaustoimintojen viiveet, jolloin FLOC-lohkon toimintaviive tulee asetella suojaustoimintojen aikaviiveitä pienemmäksi. /2/

Kolmas vaihtoehto on jatkuva liipaisu, jossa FLOC-lohkon ulostulonastat päivitetään tietyin aikavälein. /2/

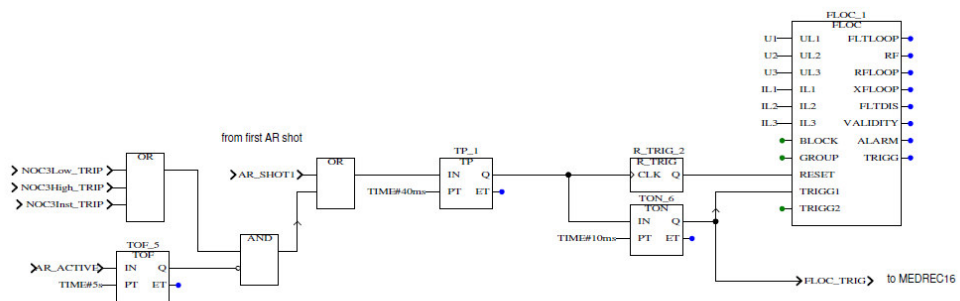
FLOC-lohkoa käytettäessä tulee huomioida lohkon kolme kirjoitusrekisteriä, joita käytetään syklisesti kiertämällä. Vian tapahtuessa vika-arvot kirjataan lohkon muistirekisteriin. Kun jollekin vialle poimitaan tiedot toimilohkolta, voivat arvot olla mitkä tahansa kolmesta. Käytännössä tämä ongelma saadaan ratkaistua käyttämällä FLOC-lohkon resetointia tietyllä viiveellä, jolloin käytetään ainoastaan yhtä muistirekisteriä. Lisäksi voidaan kaikkien kolmen eri muistirekisterien vikatietojen aikaleimoja verrata keskenään ja poimia viimeisimmän tapahtuman vikatiedot. Relekonfiguraatioon tehtävällä muutoksella saadaan suoritettua resetointi sivulla 24 olevien kuvien 9-11 mukaisesti. /10/



**Kuva 9.** AR5Func (jälleenkytkentä) toimilohkoon tehtävät muutokset FLOC lohkon resetointia varten.



**Kuva 10.** MMIALARM-lohkon resetointi.



**Kuva 11.** FLOC-lohkon resetointi tietyin aikavälein edellä esitetyn AR5Func lohkon, sekä suojauslohkojen toiminnan perusteella.

FLOC-lohkosta lähetettävät tiedot tulevat SPA-koodeina. Koodi koostuu toimilohkon numerosta, datan tyyppistä ja parametrin yksilöllisestä numerosta. SPA-koodeja tarvitaan signaalien kyselyissä sekä tietoliikenteen ristiviittausten luontiin ala-aseamalla. SPA-väylän tuominen valvomoon asti on myös mahdollista, jolloin MicroSCADAn PC-NETtiin luodun aseman alle määritetään kyseiset koodit. /5/

[58V203]	Fault impedance measurement loop
[58V204]	Fault resistance in primary ohms
[58V205]	Fault loop resistance in primary ohms
[58V206]	Fault loop reactance in primary ohms
[58V208]	Validity indicator for earth faults
[58V214]	Phase current IL1 during fault
[58V215]	Phase current IL2 during fault
[58V216]	Phase current IL3 during fault
[58V220]	Pre-fault phase current IL1
[58V221]	Pre-fault phase current IL2
[58V222]	Pre-fault phase current IL3
[58O001]	Fault impedance measurement loop
[58O002]	Fault resistance in primary ohms
[58O003]	Fault loop resistance in primary ohms
[58O004]	Fault loop reactance in primary ohms
[58O005]	Fault distance
[58O006]	Validity indicator for earth faults

### 3.3 Vikatiedot REX 521 suojuhareleessä

REX 521 suojuhareleen peruskonfiguraatio ei sisällä FLOC-lohkoa, jolloin vianpaikannus täytyy toteuttaa releestä saatavien mittaus- ja suojaustietojen perusteella, jotka ovat pääpiirteittäin samanlaiset, kuin 500-sarjan REF releissä. FLOC-lohkon puuttuminen tarkoittaa käytännössä, että vianpaikannus täytyy suorittaa vika- ja kuormitusvirtoihin, kytkinlaitteen loogiseen nimeen sekä aikatietoihin perustuen.

Rele tukee yleisimmät tiedonsiirtoprotokollat, kuten SPA, LON, MODBUS, DNP 3.0 ja IEC 60870-5-103 väylät. Vianpaikannuksen kannalta käytettävä protokolla vaikuttaa ala-aseman konfiguraation tekoon. Tässä työssä selvitetään lähinnä SPA- ja IEC- väylien käyttöä, joista kerrotaan lisää ala-aseman konfiguroinnissa.

Suojaustoimintoja REX 521 suojuhareleessä ovat:

- 3I>, 3I>> ja 3I>>>, jotka ovat kaksi- tai kolmivaiheisen suuntaamattoman oikosulun ja ylivirran suojujaukseen.
- Io>, Io>>, Io>>>, jotka ovat suuntaamattomaan maasulkusuojaukseen.
- Io>->, Io>>->, Io>>>->, jotka ovat suunnattuun ja suuntaamattomaan maasulkusuojaukseen.
- 3I>->, 3I>>->, >>>, jotka ovat kaksi- tai kolmivaiheisen suunnattuun oikosulun ja ylivirran suojujaukseen.
- 3U>, 3U>>, jotka ovat yksi, kaksi- ja kolmivaiheista ylijännitesuojausta varten.
- 3U<, 3U<<, jotka ovat yksi, kaksi- ja kolmivaiheista alijännitesuojausta varten.
- O->I, joka on jälleenkytkentätoimilohko.
- 3Ith, joka on terminen ylikuormitussuoja.

- $U_{o>}$ ,  $U_{o>>}$ ,  $U_{o>>>}$ , jotka ovat summajännitesuojaustoimilohkoja maasulkusuojaukseen.
- $F1$  ja  $f1$ , jotka ovat yli- ja alitaajuussuojaustoimilohkot.

Suojaustoiminnon perusteella vikatilanteessa saadaan pääteltyä vian tyyppi. Tätä tietoa tarvitaan vianpaikannuksessa.

Vianpaikannuksen kannalta merkittäviä mittaustoimilohkoja REX 521 releessä ovat:

- $3I$ , joka mittaa todelliset teholliset vaihevirrät.
- $I_o$ , joka mittaa tehollisen summavirran.
- $U_o$ , joka mittaa tehollisen summajännitteen.
- $3U$ , joka mittaa tehollisen vaihe- tai pääjännitteen
- $PQE$ , joka mittaa perustaajuuden kolmivaiheista tehoa ja energiaa.

REX 521 on esimerkki releestä, jossa ei ole FLOC-vianpaikannustoimilohkoa. Tällaisten releiden tapauksessa vianpaikannustiedot on otettava suoraan releen suojaustoiminnoista, jolloin tarvitaan erillisiä komentoproseduureja, joita ajetaan lopullisen laukaisun ohjaamina. Proseduureihin täytyy rakentaa myös vertailu aikaleimoille, jolloin saadaan suojauslohkon muistirekisterin viimeisin arvo. Lisäksi kuormitusvirrat ennen vikaa ovat tarvittavia tietyissä tilanteissa, esimerkiksi silloin kun johtolähdöillä on vanhoja suojareleitä ja vikavirran mittaus on otettu suoraan pääkatkaisijan suojareleelta. Näin voidaan huomioida kuormitusvirroista syntyvät muutokset vianpaikannukseen. /9/

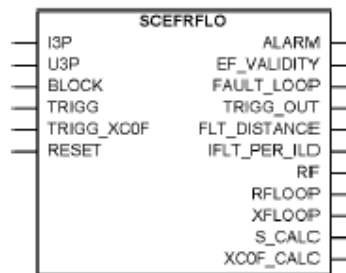
### **3.4 Vikatiedot REF6xx suojareleessä**

REF600-sarja on ABB:n uusimpia relesarjoja, joka sisältää kattavat mittaus- ja suojaustoiminnot. Vianpaikannukseen liittyvät toiminnot ovat pääosin samat, kuin 500-sarjan REF-suojareleissä. Suurimpana muutoksena ovat toimilohkojen nimet, konfigurointi-alusta sekä tuetut protokollat.

600-sarjan releet tukevat yleisimmät protokollat, kuten uuden IEC 61850 protokollan. Releissä on mahdollisuus myös GOOSE-toiminnolle, joka mahdollistaa reletason horisontaalisen liikennöinnin. Näin saadaan releiden välille luotua älyä, joka mahdollistaa nopeammat reagoinnit erilaisiin vikatilanteisiin. Lisäksi REF630 sisältää päivitetyn vianpaikannuslohkon (SCEFRFLO). /4/

### 3.4.1 REF630 Fault location toimilohko

REF630 suojareleen vianpaikannustoimilohko on pääpiirteittäin samanlainen, kuin 500-sarjalaisen FLOC-lohko. Nimeltään lohko on SCEFRFLO. Suurimpana uudistuksena vanhaan FLOC-lohkoon verrattuna on virta- ja jännitetietojen syöttö. Molemmat arvot voidaan syöttää suoraan I3P ja U3P arvoina, eli kolmivaiheisena tehollisarvona. Lohkossa on myös enemmän ulostuloja kuin perinteisessä FLOC-lohkossa.



**Kuva 12.** 600-sarjan vianpaikannuslohko.

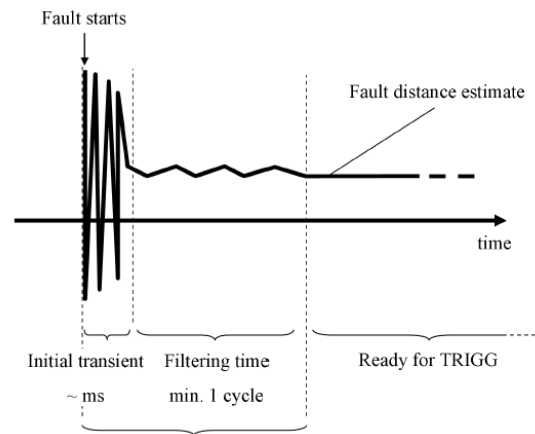
Name	Type	Default	Description
I3P	GROUP SIGNAL	-	Three phase group signal for current inputs
U3P	GROUP SIGNAL	-	Three phase group signal for voltage inputs
BLOCK	BOOLEAN	0	Signal for blocking the triggering
TRIGG	BOOLEAN	0	Triggering signal for outputs of the function
TRIGG_XC0F	BOOLEAN	0	Triggering signal for XC0F calculation
RESET	BOOLEAN	0	Input signal for resetting registers

Name	Type	Description
ALARM	BOOLEAN	Alarm signal indicating that fault is located between set limits
EF_VALIDITY	INTEGER	Validity of earth fault location
FAULT_LOOP	INTEGER	Fault impedance loop used for distance measurement
TRIGG_OUT	BOOLEAN	Signal indicating function triggering
FLT_DISTANCE	REAL	Fault distance in units selected by the user (pu)
IFLT_PER_ILD	REAL	Ratio between fault current and load current in case of an earth fault
RF	REAL	Fault resistance in primary ohms
RFLOOP	REAL	Fault loop resistance in primary ohms
XFLOOP	REAL	Fault loop reactance in primary ohms
S_CALC	REAL	Estimated equivalent load distance
XCOF_CALC	REAL	Estimated phase-to-earth capacitance of line

**Kuva 13.** SCEFRFLO-lohkon input ja output-nastojen selitykset.

SCEFRFLO-lohkokossa sisäiset toiminnot ovat pääpiirteittäin samanlaiset kuin 500-sarjan FLOC-lohkokossa. Suoritettujen suojausten perusteella päätellään vian tyyppi ja valitaan oikea laskentamalli riippuen onko vika oiko- vai maasulku. Maasulun paikannukseen liittyvät toiminnot ovat laajentuneet verrattuna entiseen FLOC-lohkokoon. Lisäksi on erillinen hälytys ulostulo, sekä S\_CALC toiminto, jolla saadaan paikallistettua johtolähdöllä oleva suurin kulutus piste.

Lohkon liipaisu voidaan myös suorittaa kolmella eri tapaa, kuten 500-sarjan FLOCissa. Liipaisuissa on huomioitava, että lohko tarvitsee vähintään kaksi sykliä, suoriutuakseen vianpaikannuslaskennoista luotettavasti. Lohkoa ei voi käyttää niin, että vian syntyessä havahtuminen, laskenta ja liipaisu tapahtuisivat yhtä aikaa. Block-nastan avulla saadaan viivästettyä vianpaikannustiedon lähettämistä niin, että laskennat ehtivät tarkentua kahden syklin ajan. /4/

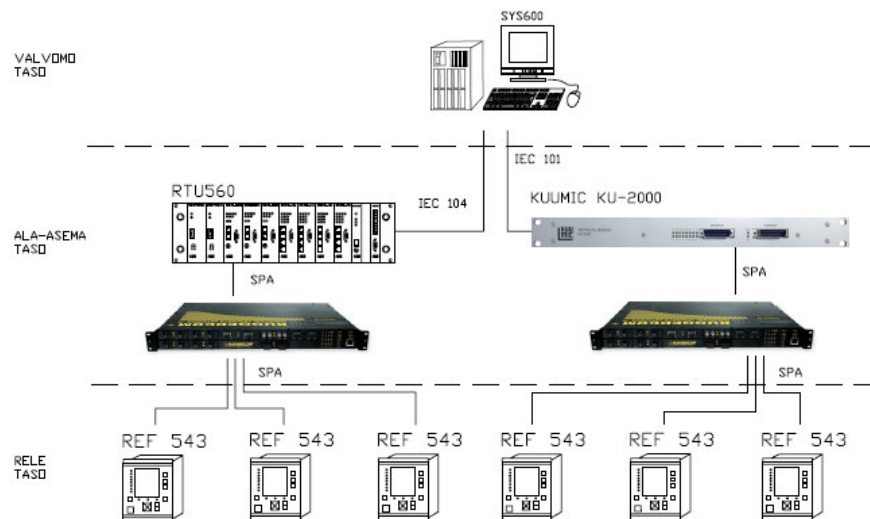


**Kuva 14.** Viassa ilmenevien transienttien huomiointi SCEFRFLO-lohkon liipaisussa.

## 4 VIANPAIKANNUS ALA-ASEMATASOLLA

### 4.1 Yleistä ala-asematason kommunikaatiosta

Sähkönjakelun automaatiossa ala-asemalla tarkoitetaan releen ja valvomon välissä sijaitsevaa solmupistettä. Ala-asema kommunikoi alaspäin suojuareille, sekä ylöspäin valvomoon. Kommunikaatio voi olla myös ala-asemien välistä, esimerkiksi RTU master – slave tyyppistä. Tyypillisesti ala-asema toimii protokolla muuntimena, joka muuntaa reletason protokollat valvomotason protokolliksi. ABB:n järjestelmissä releiltä ala-asemalle kommunikaatio on yleisimmin SPA- tai IEC protokollaa ja ala-asemalta ylöspäin IEC 60870-5-101 tai -104 protokollalla.



**Kuva 15.** Esimerkkikuva RTU ja Kuumic tyyppisten ala-asemien sijoittumisesta järjestelmässä.

Käytännössä ala-aseman tyyppi ja siihen valittavat liikennöintikortit ovat aina tapauskohtaisia. Tässä työssä ala-asema tason liikennöintiä tarkastellaan vianpaikannustietojen siirtoon vaikuttavien tekijöiden osalta.

Tarkasteltavia ala-asemia ovat Kuumic KU-2000, sekä ABB:n COM 600 ala-asema. Molemmat ovat toimintaperiaatteeltaan protokollamuuntimia. Kuumic on hieman yksinkertaistettu ala-asema, kun taas COM 600 on toiminnoiltaan moni-

puolisempi. Monissa järjestelmissä on usein erilaisia ala-asetatyyppjejä, joita voidaan käyttää yhtäaikaaisesti. /15/

#### 4.2 Vikapaketin signaalien ristiviittaus Kuumic KU-2000 ala-asetalla

Kuumic ala-asetan peruskonfiguraatio muodostuu liikennöintiporttien toiminnan määrittelystä, pääohjelmasta sekä parametritiedosta. Itse signaalien käsittelyyn vaikutetaan erillisellä tekstitiedostolla, jossa määritellään ristiviittaukset protokollien välillä.

Peruskonfiguraation pohjalta ala-asetaan luodaan määrittelyt käytettäville protokollille ja liikennöinnille. Oheisessa kuvassa (kuva 16) on esitettynä Rku4win käyttöliittymän peruskuvaa, josta ilmenevät ladatun pääohjelman ja parametritiedoston versiot sekä määritellyt tiedot protokollille ja liikennöinnille.

```

Local operation
87654321
Iec line 0 monitor      DIP 1 = 00011011
                        DIP 2 = 00011011
iec line 1 monitor
IEC buffers
Spa line 1 monitor      Port 0, Pri 0 = 0022, Pri 1 = 0000 Byt
                        Port 1, Pri 0 = 0000, Pri 1 = 0000 Byt
sPa line 2 monitor
Unfreeze monitor buffers
memory Dump

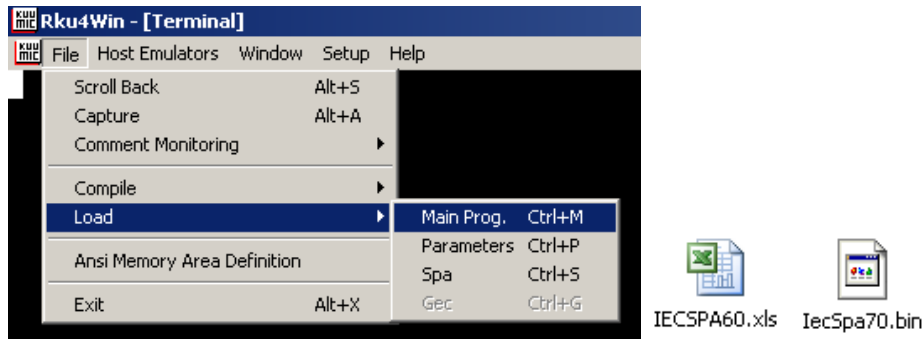
Parameters : IECSPA60.txt 32401 01.03.10 11:29
Spa Config !!! Bad !!!

Kuumic Oy (c) 2007, IEC870-5 / SPA Protocol Converter Vers. 7.0

```

**Kuva 16.** Rku4win käyttöliittymän pääkuva.

Mikäli ala-asetan peruskonfiguraatiota täytyy muuttaa, onnistuu se lataamalla uusi pääohjelma, joka on .bin muotoinen tiedosto sekä muuntamalla parametri tiedosto .xls muodosta .txt muotoon ja lataamalla sekin alas.



**Kuva 17.** Pääohjelman ja parametrien lataus sekä ladattavat tiedostot.

Signaalilistasta käytetään nimitystä SPA-kuvaus. Kuvaus tehdään tekstieditorilla, tai excelillä lopuksi kääntäen tekstitiedostoksi.

SPA-kuvaus aloitetaan tekemällä yleiskyselyosuus (GI, general interrogation), jossa määritellään yhteyskatkon jälkeen releeltä kysyttävät signaalit. Katkoksen jälkeen annetaan yleiskyselykäsky, joka käynnistää kyselyn. Aloitus voidaan määrittää johonkin tiettyyn osoitteeseen. Yleensä kyselyn aloitus määritetään IEC osoitteesta C\_IC\_NA\_1 X, joka on interrogation käsky. X kirjaimen tilalle määritellään mielivaltainen IEC-osoite. Kyselyn käynnistyessä määriteltyihin SPA-osoitteisiin lähetetään kysely. Kuvassa 18 näkyy yhden linjan alle tehty kysely. Osuudessa ”TO SPA: >31R120V1” on määritetty tietty signaali, joka halutaan kysyä releeltä. Numero 31 kertoo kyseltävän releen orjanumeron, joka on vastaava, kuin unit number. R kertoo pisteen olevan read muotoa, eli kyseistä pistettä halutaan lukea. /15/

Yleiskyselyn jälkeen tehdään ristiviittaukset jokaiselle suojareleelle / linjalle erikseen. Tässä vaiheessa jokainen signaali on ristiviitattava SPA signaalista IEC signaaliksi. Käytännössä haetaan releen parametri- ja tapahtumalistauksista kullekin signaalille oma osoite, jonka jälkeen viitataan signaali IEC muotoon. Spa-osoite kirjoitetaan esimerkiksi muotoon ”FROM SPA: <31D31E2/3” ja IEC viittaus muotoon ”TO IEC: M\_SP\_TA\_1 3200” 0/1. Iec osoitteeseen tuleva kirjain-alaviivayhdistelmä kertoo datan tyyppin ja numerot kertovat Iec osoitteen. Kuvassa 18 on esitettynä SPA-kuvauksen luonti vikatiedoille. /15/

```

Kuumic_FLOC.txt - Notepad
File Edit Format View Help
;Vianpaikannus
FROM SPA: <31R5801/6          TO IEC: M_ME_NC_1 306          ;vikatiedot FLOC:lta (allapuolella selitys)
;31R5801 306 vaihetieto FLOC:lta
;31R5802 307 vikaresistanssi FLOC:lta (ohm)
;31R5803 308 vikaresistanssi (LOOP) FLOC:lta (ohm)
;31R5804 309 vikaresistanssi FLOC:lta
;31R5805 310 vian etäisyys FLOC:lta
;31R5806 311 Maasulun paikannuksen tarkkuus
;vikavirrat FLOC:lta
;kuormitusvirrat FLOC:lta

FROM SPA: <31R58V214/216      TO IEC: M_ME_NC_1 312
FROM SPA: <31R58V220/222      TO IEC: M_ME_NC_1 315

;Jälleenkytkennät
FROM SPA: <31R8054          TO IEC: M_SP_NA_1 334 0/1      ;JK päällä/pois
FROM SPA: <31D80E34/35      TO IEC: M_SP_TA_1 334 0/1

FROM SPA: <31R82E0/1        TO IEC: M_SP_NA_1 335 0/1      ;AJK käy

FROM SPA: <31R81E3          TO IEC: M_SP_TA_1 336 1        ;PJK I>
FROM SPA: <31R81E2          TO IEC: M_SP_TA_1 337 1        ;PJK I->
FROM SPA: <31R81E4          TO IEC: M_SP_TA_1 338 1        ;PJK I0-> >
FROM SPA: <31R81E5          TO IEC: M_SP_TA_1 339 1        ;PJK I0-> >>
FROM SPA: <31R82E3          TO IEC: M_SP_TA_1 340 1        ;AJK I>
FROM SPA: <31R82E2          TO IEC: M_SP_TA_1 341 1        ;AJK I->
FROM SPA: <31R82E4          TO IEC: M_SP_TA_1 342 1        ;AJK I0-> >
FROM SPA: <31R82E5          TO IEC: M_SP_TA_1 343 1        ;AJK I0-> >>
FROM SPA: <31R81E6          TO IEC: M_SP_TA_1 318 1
FROM SPA: <31R82E6          TO IEC: M_SP_TA_1 319 1
;onnistunut PJK
;onnistunut AJK

FROM SPA: <31R80E7          TO IEC: M_SP_TA_1 344 1
TO SPA: >31R5801/6
TO SPA: >31R58V214/216
TO SPA: >31R58V220/222
TO SPA: >31W135V1:0:
;Lopullinen laukaisu
;vikatietojen luku FLOC:lta
;vikavirtojen luku
;kuormitusvirtojen luku
;JK:t pois lopullisen laukaisun jälkeen
;Lopullinen laukaisu

FROM SPA: <31R80E6          TO IEC: M_SP_TA_1 344 0

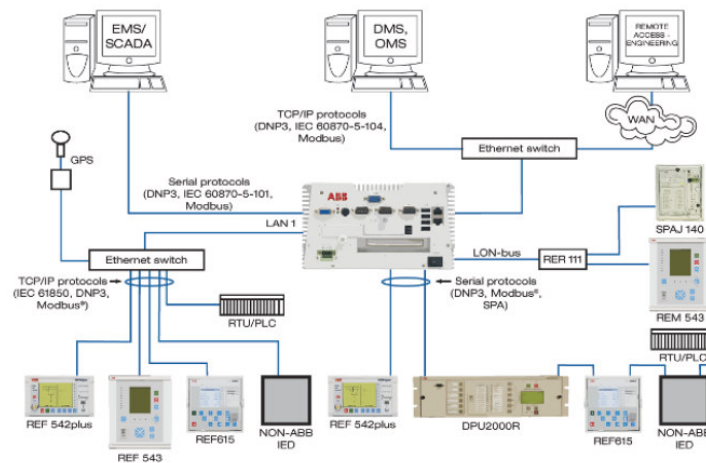
```

**Kuva 18.** Vianpaikannukseen tehtävät ristiviittaukset KU-2000 ala-asemalla.

Vianpaikannuksen näkökulmasta huomiota tulee kiinnittää releeltä saataviin tietoihin. Mikäli releen konfiguraatiossa ei ole FLOC- tai SCEFRFLO-lohkoa, ei ala-asema konfiguraatioon ole järkevää tehdä ristiviittauksia vianpaikannuslohkon signaalien kuljettamiseen. Mikäli kyseessä on johtolähtö, jonka releistystä ollaan tulevaisuudessa uusimassa, voi vianpaikannuslohkon signaalien luonti ja jättäminen kommentoiduksi tulevaisuutta varten olla hyödyllistä. Tilanteessa, jossa releen konfiguraatio ei sisällä FLOC- tai SCEFRFLO-lohkoa, mutta konfiguraatiossa olevat suojauslohkot kykenevät antamaan muistirekisteristä vikavirtatiedot, tulee kyseiset signaalit ristiviitata. /15/

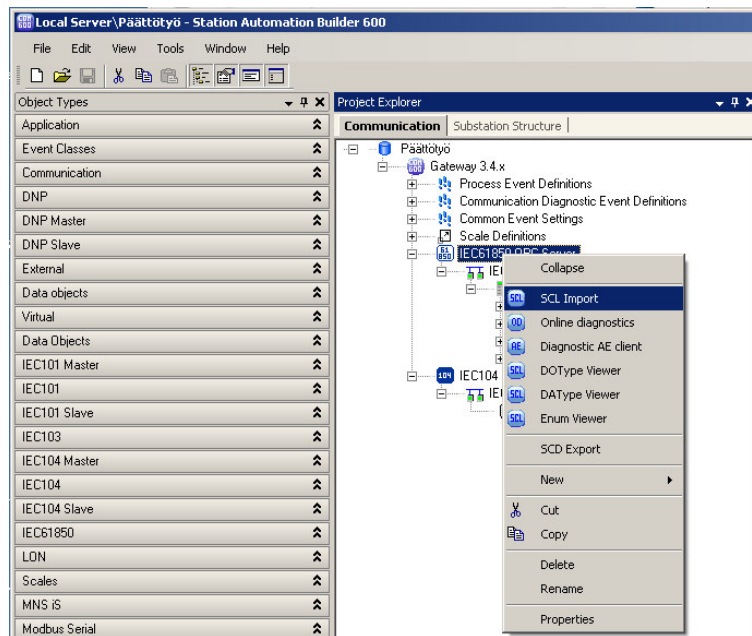
### 4.3 Signaalien ristiviittaus COM 600 ala-asemalla

COM 600 on ala-asema, joka kykenee linkittämään sähkönjakelun automaation laajalti yhdeksi kokonaisuudeksi lukuisilla eri protokollilla. COM 600 tyyppisessä ala-asemassa suoritetaan samantyyppiset ristiviittaustoiminnot, kuin Kuumic KU-2000 ala-asemassa. Tässä työssä perehdytään erityisesti ala-asemalta tulevien vianpaikannukseen liittyvien signaalien ristiviittaukseen. Työssä toteutetaan esimerkki, jossa relepuolen kommunikaatio on IEC 61850- protokollaa ja valvomo- puolen kommunikaatio IEC-104- protokollaa. Ristiviittaustaulukko tehdään REF630 suojausloheen vianpaikannuslohkon (SCEFRFLO) signaaleille, jotka tulevat releeltä IEC 61850 standardin mukaisesti nimettyinä. /13/



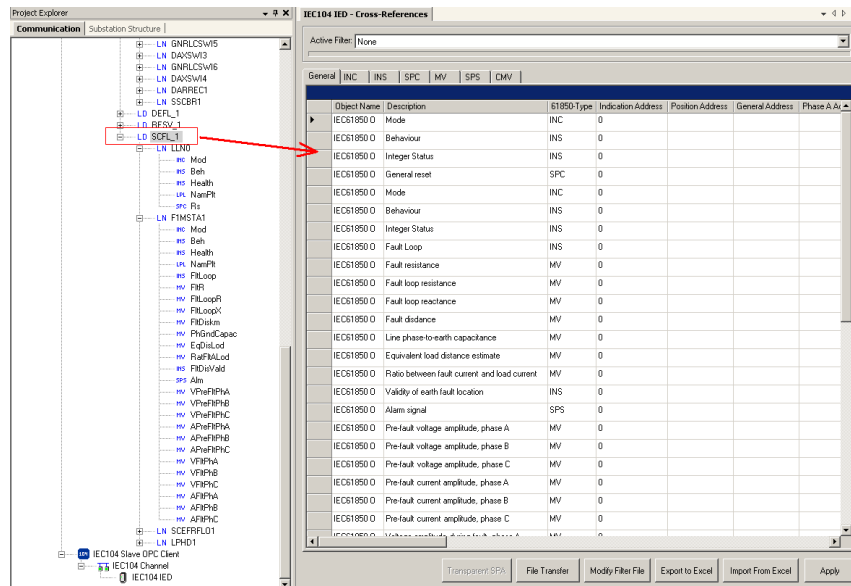
**Kuva 19.** COM 600 ala-aseman sijoittuminen järjestelmässä.

Luotaessa COM 600 konfiguraatiota, lähtee prosessi käyntiin relekonfiguraation .ar- tiedostosta. Kyseinen tiedosto sisältää kaikki releeseen luodut toimilohkot ja niistä saatavat signaalit. Tiedoston pohjalta luodaan CET ohjelman (Communication engineering tool) avulla .cid- tiedosto. Cid- tiedoston avulla saadaan tuotua signaalit SAB 600 ohjelmaan. SAB 600 ohjelmalla saadaan luotua puutyypinen projektiluonnos, jonne voidaan lisätä gateway- tyyppi, OPC server-client rajapinta sekä käytettävät protokollat. Protokollavalintojen alle voidaan luoda ”IED”, joka tarkoittaa relettä. IED:lle voidaan suorittaa SCL import, jonka avulla voidaan luodun .cid- tiedoston sisältö tuoda puurakenteeseen mukaan. Käytännössä tämä tarkoittaa releen signaalien importtausta SAB 600- ohjelmaan. Cid- tiedosto tuodaan OPC serverin puolelle, eli tässä tapauksessa IEC 61850 IED:n alle, koska signaalit ovat kyseisen releen konfiguraatiota. /13/



**Kuva 20.** 61850 OPC serverille tehtävä SCL Import SAB 600 ohjelmassa.

SCL Importin jälkeen saadaan kaikki signaalit puurakenteeseen. Seuraavaksi luodaan ristiviittaukset OPC clientin alla olevalle IEC 60870-5-104 IED:lle. Tässä tapauksessa IEC 60870-5-104 IED:llä tarkoitetaan valvomon pään tietokonetta. Cross-reference taulukkoon saadaan siirrettyä kaikki signaalit, jotka halutaan ristiviitata valvomotason IEC 60870-5-104 protokollalle. Tässä työssä olevassa esimerkissä on ristiviittaus taulukkoon poimittu REF630 suojarahkeen SCEFRFLO-toimilohkon signaalit. /13/



**Kuva 21.** SCFL (600-sarjan vianpaikannus) lohkoista pimitut signaalit

Ristiviittaustaulukossa signaaleille annetaan yksilölliset Indication Addressit, joiden avulla saadaan kukin signaali linkitettyä MicroSCADAn prosessipisteiden OA- attribuutteihin. Eri tyyppin signaalit näkyvät kukin omalla välilehdellään.

Object Name	Description	61850-Type	Indication Address	Priority	Class	InterrogationGroup	UpdateRate	OverWrite	TimeTagHandling	SendMeasureAndValType
IEC61850 0	Fault resist	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Fault loop r	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Fault loop r	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Fault dista	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Line phase	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Equivalent	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Ratio betw	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Pre-fault vo	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Pre-fault vo	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Pre-fault vo	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Pre-fault cu	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Pre-fault cu	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Pre-fault cu	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Voltage am	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Voltage am	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Voltage am	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Current am	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Current am	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Current am	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Fault Dista	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Fault loop r	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Fault loop r	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Line phase	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Ratio betw	MV	0	0	2	1	1000	True	0	0
IEC61850 0	Equivalent	MV	0	0	2	1	1000	True	0	0

**Kuva 22.** Signaaleille annettavat osoitteet IEC 60870-5-104 kommunikaatiota varten.

Luotaessa ristiviittaukset ala-asema koniguraatioon tulevat signaalit automaattisesti oikeiksi, kun .cid tiedosto luodaan relekonfiguraation perusteella CET-ohjelmalla. Mikäli releen konfiguraatiossa ei ole FLOC- tai SCEFRFLO-lohkoa, siirrettävät vikatiedot ovat vain virta ja joissain tapauksissa myös jännite. /13/

## 5 VIKATIEDOT KÄYTÖNVALVONTAJÄRJESTELMÄSSÄ

### 5.1 Yleistä vikatietojen keruusta käytönvalvontajärjestelmään

Käytönvalvontajärjestelmä kuuluu valvomotason automaatioon, johon sisältyvät kytkinlaitteiden tilatiedot ja ohjaukset, mittaukset, hälytykset sekä raportointi. Tässä työssä käsitellään ABB:n kehittämää MicroSCADA Pro Ver. 9.2 käytönvalvontajärjestelmää.

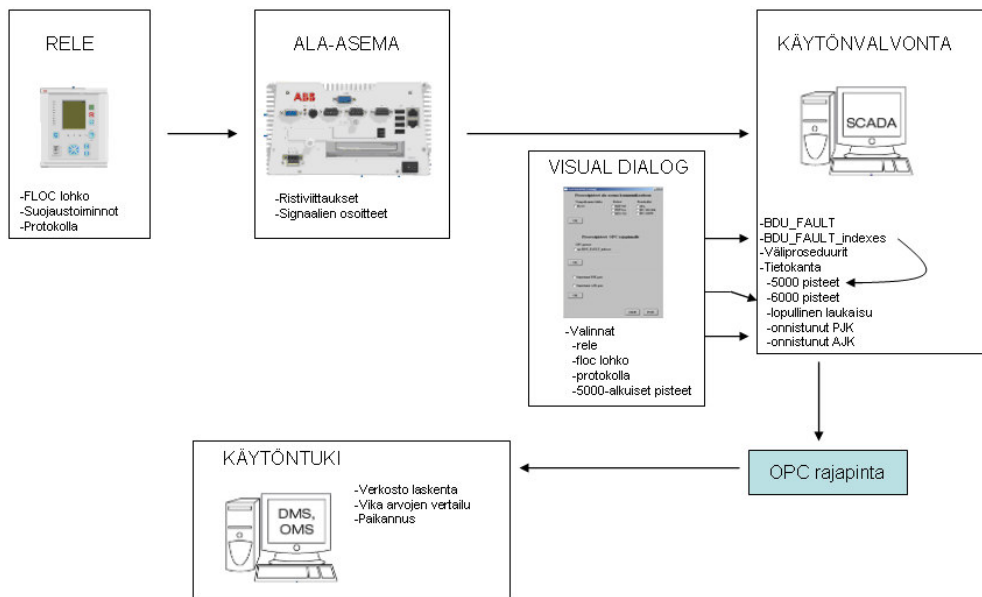
Lähtökohtaisesti optimitilanteessa käytönvalvontajärjestelmä ei puutu vikatietojen käsittelyyn muuten kuin välittämällä releeltä tai ala-asemalta tulleen vikapaketin eteenpäin käytöntukijärjestelmälle. Vanhemmissa järjestelmissä, joissa releiltä joudutaan keräämään erikseen vikaan liittyvä data, joudutaan tekemään GI tyyllisiä ratkaisuja, sekä syvempää älyä komentoproseduureihin. Uudemmissa järjestelmissä vikapaketti muodostetaan MicroSCADAssa käytettävien prosessipisteiden ja komentoproseduurien avulla.

Haasteellisuuden vikapaketien keruussa ja siirrossa valvomotasolla luovat lukuisat eri rele- ja ala-asema tyypit, jotka vaikuttavat tiedon keräämisen tapaan. Lisäksi käytettävät protokollat muuttavat tiedonkeruuta. Vikatiedot voidaan kerätä GI tyyllisesti, jolloin vian sattuessa lähetetään SPA-kysely, jossa kysytään tietyt vikatietoihin liittyvät parametrit ja jäädään odottamaan vastausta releeltä. Tällaisessa tapauksessa releeltä tulevan vastauksen parametritiedot kirjoitetaan suoraan BDU\_FAULT:C- proseduriin, joka lähettää tiedot eteenpäin käytöntukijärjestelmälle.

Toinen vaihtoehto on käyttää BDU\_FAULT\_indexes:C proseduuria, joka luo automaattisesti vianpaikannukseen tarvittavat prosessipisteet, joiden tiedot lähetetään käytöntukijärjestelmälle edellä mainitun BDU\_FAULT:C proseduurin avulla. Tällaisessa menetelmässä konfiguraation tekevän henkilön täytyy ainoastaan ajaa proseduri, joka luo prosessipisteet ala-asemalta tulevia signaaleita varten, tai luoda käsin tarvittavat prosessipisteet sekä luoda erillinen proseduri, joka käynnistää ja siirtää tiedot BDU\_FAULT:C proseduurin. Tulevaisuuden kannalta paras ratkaisu on jälkimmäisenä esitetty prosessipisteisiin pohjautuva ratkaisu, koska se

on selkeämpi ja helpompi luoda. Kuitenkin käytettävät releet ja protokollat asettavat tiettyjä rajoitteita siirtoproseduurien käytössä. Esimerkiksi toimitukset, joihin sisältyy muitakin kuin ABB:n releitä, vaikeuttaa prosessipisteisiin perustuvan vikapakettien siirtomenetelmän käyttöä. Edellä mainitut proseduurit käydään läpi seikkaperäisesti työn myöhemmässä vaiheessa. /10/

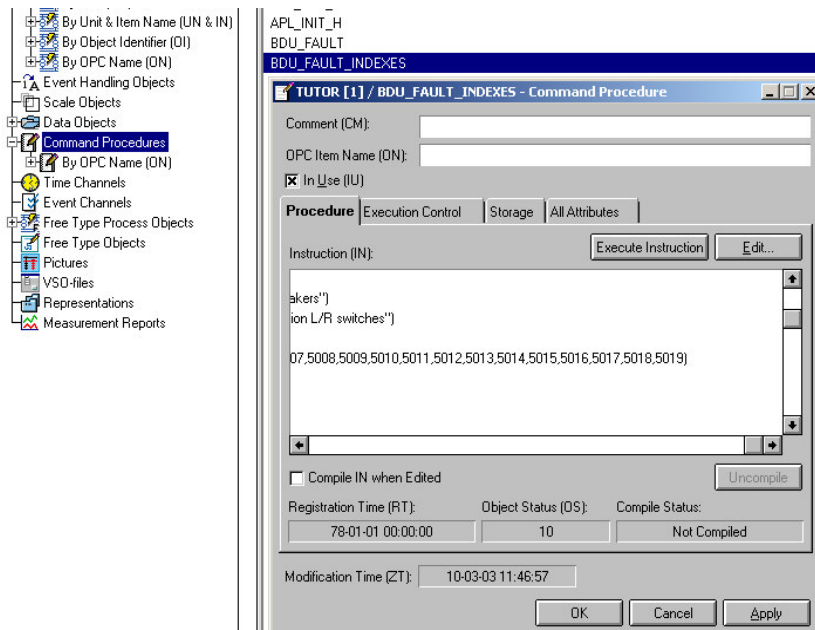
Tässä työssä esitettävät ratkaisut perustuvat prosessipisteiden avulla suoritettavaan vikapakettien siirtoon. Muut siirtotavat käydään läpi pintapuolisesti.



**Kuva 23.** Kokonaiskuvaus luodusta vikapakettien keruusta ja siirrosta.

## 5.2 Komentoproseduurit

MicroSCADAssa komentoproseduurilla tarkoitetaan SCIL-ohjelmoitua komentoketjua, joka suoritetaan haluttujen ehtojen täytyttyä. Vikapakettien siirrossa käytetään BDU\_FAULT:C proseduuria, joka siirtää vikapakettien tiedot SYS 600 käytönvalvontajärjestelmästä DMS 600 käytöntukijärjestelmään sekä vikatiedonsiirron ollessa prosessipisteisiin perustuvaa, käytetään lisäksi BDU\_FAULT\_indexes:C proseduuria.



**Kuva 24.** BDU\_FAULT\_indexes:C proseduurin suoritus prosessipisteiden luontia varten.

BDU\_FAULT:C proseduuria voidaan käyttää SCIL-API tai OPC rajapinnalla. SCIL-API on siirtotapa, jossa vikatiedot siirretään muuttujajoukkona. SCIL-API:n muuttujajoukko koostuu komentoproseduurissa kerätyistä vektoreista. OPC on näistä kahdesta uudempi tiedonsiirtotapa. Prosessipistepohjainen ratkaisu on tehty OPC rajapintaa varten. BDU\_FAULT:C proseduurin tehtävä on siirtää vikapaketti OPC rajapinnalle. Proseduuri kirjoittaa vikatiedot aina indeksiltään 5000-alkuisiin prosessipisteisiin, jotka saadaan luotua BDU\_FAULT\_indexes proseduurilla.

LN	BK	UN	[DA]IN	[DB]EH	OI	OX
S1_J01_Q1		13			S1_J01_Q1	Breaker command
S1_J01_Q1		15			S1_J01_Q1	Breaker device control block
S1_J01_Q1		16			S1_J01_Q1	Breaker open interlocking
S1_J01_Q1		17			S1_J01_Q1	Breaker close interlocking
S1_J01_Q1		18			S1_J01_Q1	Cause of interlocking
S1_J01_Q1		19			S1_J01_Q1	Breaker selection on monitor
S1_J01_Q1		20			S1_J01_Q1	Breaker command event
S1_J01_Q1		113			S1_J01_Q1	Breaker command
S1_J01_Q1		5000			S1_J01_Q1	DMS OPC: Trigger
S1_J01_Q1		5001			S1_J01_Q1	DMS OPC: Fault Type
S1_J01_Q1		5002			S1_J01_Q1	DMS OPC: Phase
S1_J01_Q1		5003			S1_J01_Q1	DMS OPC: Fault Current
S1_J01_Q1		5004			S1_J01_Q1	DMS OPC: Load Current before the Fault
S1_J01_Q1		5005			S1_J01_Q1	DMS OPC: Zero Current
S1_J01_Q1		5006			S1_J01_Q1	DMS OPC: Zero Voltage
S1_J01_Q1		5007			S1_J01_Q1	DMS OPC: Earth Fault Distance
S1_J01_Q1		5008			S1_J01_Q1	DMS OPC: Earth Fault Distance Z
S1_J01_Q1		5009			S1_J01_Q1	DMS OPC: Transformer P
S1_J01_Q1		5010			S1_J01_Q1	DMS OPC: Transformer Q
S1_J01_Q1		5011			S1_J01_Q1	DMS OPC: Method Name
S1_J01_Q1		5012			S1_J01_Q1	DMS OPC: Resistance of Impedance
S1_J01_Q1		5013			S1_J01_Q1	DMS OPC: Reactance of Impedance
S1_J01_Q1		5014			S1_J01_Q1	DMS OPC: Fault Distance
S1_J01_Q1		5015			S1_J01_Q1	DMS OPC: Time of Event: RT
S1_J01_Q1		5016			S1_J01_Q1	DMS OPC: Time of Event: RIM
S1_J01_Q1		5017			S1_J01_Q1	DMS OPC: Notation of measurement
S1_J01_Q1		5018			S1_J01_Q1	DMS OPC: Source LN
S1_J01_Q1		5019			S1_J01_Q1	DMS OPC: Transformer LN
S1_J02		10			S1_J02	Bay local/remote switch
S1_J02		15			S1_J02	Ext. interlocking (hardware)
S1_J02		16			S1_J02	Ext. interlocking (software)
S1_J02		17			S1_J02	Ext. interlocking command
S1_J02		18			S1_J02	Internal interlocking
S1_J02		19			S1_J02	Internal interlocking command

**Kuva 25.** OPC rajapintaa varten luodut indeksiltään 5000-alkuiset vianpaikannus prosessipisteet.

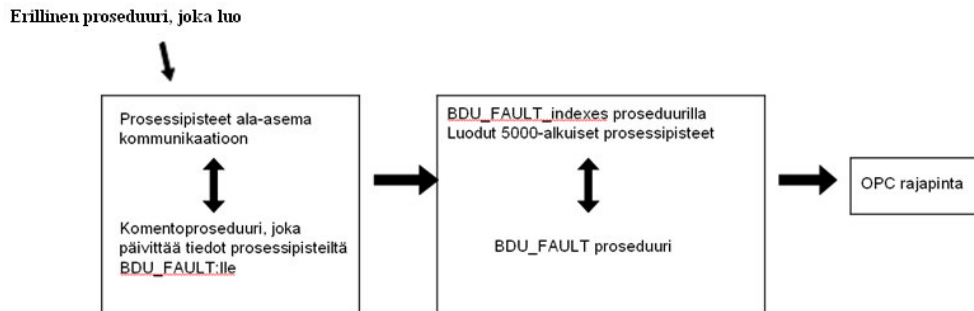
Käytettäessä OPC rajapintaa täytyy huomioida, että BDU\_FAULT:C proseduurissa on luotuna valmius sekä OPC liittynälle että SCIL-API muuttujajoukolle. Käytettäessä OPC rajapintaa täytyy SCIL-API osuus poistaa, tai jättää kommenttiksi proseduriin.

### 5.3 Prosessipisteet vianpaikannuksessa

Vianpaikannusprosessipisteet voidaan jakaa kahteen eri tasoon käytettäessä OPC rajapintaa. Pisteet muodostuvat ala-asemalle suoritettavaan kommunikointiin tarvittavista pisteistä sekä jo esitellyistä indeksiltään 5000-alkuisista prosessipisteistä.

Ala-asema kommunikaatiota varten luotavia pisteitä varten työn ohessa tehtiin erilliset proseduurit IEC 60870-5-101/-104, IEC 61850 ja SPA-protokollia varten, jotka luovat tarvittavat pisteet ala-asemakommunikaatiota varten MicroSCADAn tietokantaan. Proseduurien toiminta on kuvattuna kappaleessa 5.4. Osoitteen rakentuminen MicroSCADassa vaihtelee protokollan mukaan. Aiemmin esitettyjen indeksiltään 5000-alkuisten vianpaikannuspisteiden osoitteisiin ei tarvitse puuttua,

koska BDU\_FAULT:C luo kerralla pisteet oikean tyyppisiksi OPC rajapintaa varten.



**Kuva 26.** Toimintoketju ja prosessipisteiden käyttö MicroSCADAssa, käytettäessä OPC rajapintaa vianpaikannuksessa.

#### 5.4 Luotujen proseduurin kuvaus

Ala-aseimalta kerättäviä tietoja varten on luotava aina prosessipisteet. Prosessipisteiden tyyppi riippuu käytettävästä protokollasta sekä datan tyypistä. Vianpaikannustiedot saadaan kerättyä ”analog input” tyyppisiin prosessipisteisiin, koska kaikki luettava data on numeerisessa muodossa. Siirtoproceduuriin on rakennettu komentoja, jotka muuttavat numeeriset arvot tarvittaessa tekstimuotoon. Prosessipisteen protokollatyyppi voi taas olla miltei mikä tahansa MicroSCADAssa käytettävä protokolla. Työn ohessa luodut proseduurit ovat protokollakohtaisia, koska tietyissä tilanteissa samassa tietokannassa voi olla eri protokollan omaavia prosessipisteitä katkaisijoille. Tuetut protokollat luoduilla proseduureilla ovat IEC 60870-5-101/-104, IEC 61850 ja SPA.

Proseduuri toimii hyvin samalla tavalla kuin BDU\_FAULT\_indexes. Alussa etsitään halutun tyyppisiä katkaisijoita ja lähtöjä. Mikäli käytössä on SPA tuettu proseduur, etsitään prosessipisteitä, joiden DX attribuutti on ”N7D”, eli SPA-tyyppiä, mikä määrittää käytetyn protokollan sekä etsitään TX attribuutit, joiden tyyppi on ”Breaker position indication”, jolloin varmistutaan kyseisen pisteen olevan katkaisijan. Toinen vaihtoehto on käyttää prosessipisteen RX-attribuuttia hakukriteerinä, jolloin mahdollisesti tietokannan prosessipisteiden DX-

attribuutteihin tehtävät muutokset eivät vaikuta komentoproceduren toimintaan. Näin saadaan tunnistettua kaikki katkaisijat, joiden protokolla on SPA. Jokaiselle eri proseduurille on tehty vastaavat toiminnot, riippuen tuetusta protokollasta.

```
;ETSII KATKAISIJAT, JOTKA SPA TYYPPISIÄ
; Search using LIB 5xx and SA-LIB reserved information in DX attribute
@lb_search = application_object_list(0,"ix","A","","","DX==""N7D"" and ,-
TX==""Breaker position indication""",("OI","OX","PT","DX"))
@ls_search = application_object_list(0,"ix","A","","","DX==""N7D""and ,-
TX==""Bay local/remote-switch""",("OI","OX","PT","DX"))
```

**Kuva 27.** Vikapisteproseduurin katkaisijan prosessipisteiden haku.

Oikeiden katkaisijoiden tunnistuksen jälkeen ajetaan proseduurin osa, joka luo vektorina prosessipisteet. DX-attribuutilla määritellään protokollaksi SPA, IX-attribuutilla järjestysnumero tietokantaan ja OX-attribuutilla objektin teksti. PT-attribuutti määrittää luotavan prosessipisteen analogia inputtyypiksi. Loppuosassa luodaan loopilla sama määrä ”prosessipistepaketteja”, kuin katkaisijoita löytyi. Lisäksi console\_outputin avulla saadaan MicroSCADAn notificationikkunaan tuotua tiedot löydettyistä pisteistä, sekä luoduista pisteistä. Notification ikkunaan tuodut tiedot helpottavat huomattavasti onnistuneen proseduurin ajon havainnointia.

```

;LUO PROSESSIPISTEET VEKTORINA MUOTO SPA / ANALOG INPUT
#IF %DX=="N7D" #THEN #BLOCK
@v_dx = vector("N6", "N6", "N6", "N6", "N6", "N6", "N6", "N6", "N6")
@v_ix = vector(6000,6001,6002,6003,6004,6005,6006,6007,6008)
@v_ox = vector("vikavirta L1",-
              "vikavirta L2",-
              "vikavirta L3",-
              "kuormitusvirta L1",-
              "kuormitusvirta L2",-
              "kuormitusvirta L3",-
              "vaihetieto",-
              "reaktanssitieto",-
              "vikatyyppi")
@v_pt = vector(9,9,9,9,9,9,9,9,9)
;Process objects for Breakers
;-----
#loop_with i = 1..length(lb_search:VLN)
@ln = lb_search:VLN(%i)
@oi = lb_search:VOI(%i)
@pt = lb_search:VPT(%i)
@dx = lb_search:VDX(%i)

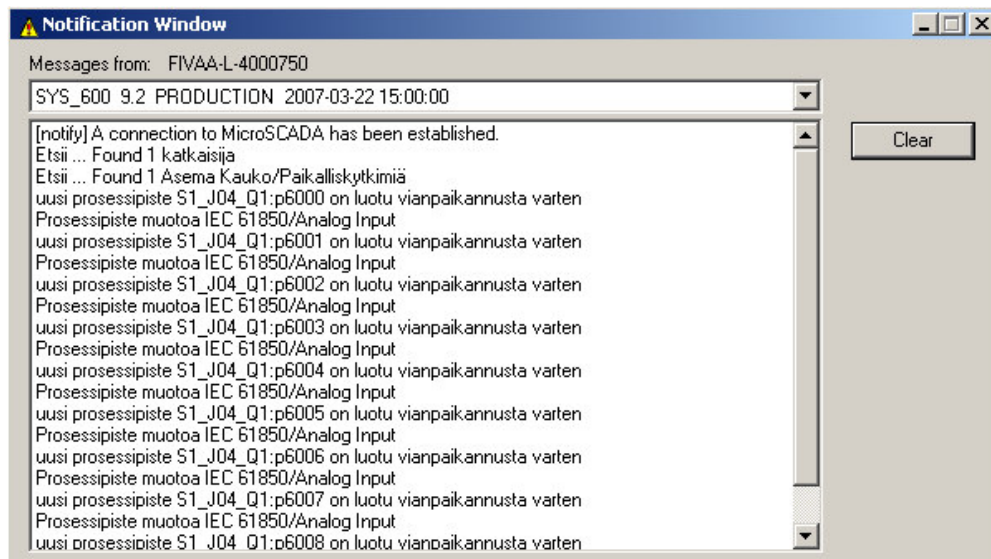
#loop_with j = 1..length(%v_ix)
@lb_Crea_Attr = list(OI = %oi,-
                   OX = "vikaprosessipiste: "+%v_ox(%j),-
                   PT = %v_pt(%j),-
                   SS = 1,-
                   DX = %v_dx(%j),-
                   IU = 1,-
                   SN = "i_1")

@ix = %v_ix(%j)
@s=0
@s=status
#error ignore
@p_created='ln':pss'ix'
@s=status
#error stop
#if %s<>0 #then #block
#create 'LN':P'IX' = %lb_Crea_Attr
@info = console_output("uusi prosessipiste 'ln':p'ix' on luotu vianpaikannusta varten")
@info = console_output("Prosessipiste muotoa SPA/Analog Input")
#block_end
#else #block
@info = console_output("prosessipiste 'ln':p'ix' on jo olemassa")
#block_end
#loop_end
;set 'ln':Pov6000 = -1.0
#loop_end

#BLOCK_END

```

**Kuva 28.** Prosessipisteiden luonti vikapisteproseduurin avulla.



**Kuva 29.** Console outputin kautta Notification ikkunaan tulostuvat tiedot löydetystä katkaisijoista sekä tietokantaan luoduista prosessipisteistä.

## 5.5 Vianpaikannuspisteiden ja BDU\_FAULT-proseduurin aktivointi

Vikatietojen päivittämiseen tarvitaan prosessipisteiden ja BDU\_FAULT komentoproceduurin lisäksi yksi erillinen väliproceduuri, joka päivittää ala-asemalta tulevien prosessipisteiden tiedot. Proseduuri siirtää ala-asemakommunikointia varten luotujen prosessipisteiden tiedot BDU\_FAULT proseduurin indeksiltään 5000-alkuisille prosessipisteille.

Proseduuria ajetaan kun joltain johtolähdöltä saadaan lopullinen laukaisu. Kyseinen tieto saadaan prosessipisteeseen, joka on linkitetty tapahtumakanavan kautta proseduriin. Prosessipisteen arvon vaihtuessa ykköseksi, alkaa suoraan proseduurin ajo. Kuvassa 31 näkyvä ”ov= =1” tarkoittaa lopullisen laukaisun prosessipisteen arvon vaihtumista ykköseksi. Tapahtumakanavan aktivointi tapahtuu, kun object value (OV) vaihtuu ykköseksi. Tapahtumakanava puolestaan aktivoi itse proseduurin. Prosessipiste aktivoi tapahtumakanavan, kun pisteelle tulee uusi arvo. Proseduurissa oleva ehto kuitenkin estää, ettei proseduuria ajeta pisteen muuttuessa nolllaksi.

Jokaiselle katkaisijalle on luotava oma prosessipiste, joka aktivoi samaa tapahtumakanavaa ja proseduuria. Edellä mainitun avulla saadaan samaan proseduriin tuotua eri katkaisijoiden lopulliset laukaisut. Kuvassa 30 on esitettyinä prosessipisteille tehtävät määrittelyt.

LN	IX	UN	[DA]/N	[OB]/EH	OI	OX
S1_J01_Q1		5002			S1_J01_Q1	DMS OPC: Phase
S1_J01_Q1		5003			S1_J01_Q1	DMS OPC: Fault Current
S1_J01_Q1		5004			S1_J01_Q1	DMS OPC: Load Current before the Fault
S1_J01_Q1		5005			S1_J01_Q1	DMS OPC: Zero Current
S1_J01_Q1		5006			S1_J01_Q1	DMS OPC: Zero Voltage
S1_J01_Q1		5007			S1_J01_Q1	DMS OPC: Earth Fault Distance
S1_J01_Q1		5008			S1_J01_Q1	DMS OPC: Earth Fault Distance Z
S1_J01_Q1		5009			S1_J01_Q1	DMS OPC: Transformer P
S1_J01_Q1		5010			S1_J01_Q1	DMS OPC: Transformer Q
S1_J01_Q1		5011			S1_J01_Q1	DMS OPC: Method Name
S1_J01_Q1		5012			S1_J01_Q1	DMS OPC: Resistance of Impedance
S1_J01_Q1		5013			S1_J01_Q1	DMS OPC: Reactance of Impedance
S1_J01_Q1		5014			S1_J01_Q1	DMS OPC: Fault Distance
S1_J01_Q1		5015			S1_J01_Q1	DMS OPC: Time of Event_RT
S1_J01_Q1		5016			S1_J01_Q1	DMS OPC: Time of Event_RM
S1_J01_Q1		5017			S1_J01_Q1	DMS OPC: Notation of measurement
S1_J01_Q1		5018			S1_J01_Q1	DMS OPC: Source LN
S1_J01_Q1		5019			S1_J01_Q1	DMS OPC: Transformer LN
S1_J01_Q1		6000	0	101	S1_J01_Q1	vik:aprosessipiste: vik:avirta L1
S1_J01_Q1		6001	0	102	S1_J01_Q1	vik:aprosessipiste: vik:avirta L2
S1_J01_Q1		6002	0	103	S1_J01_Q1	vik:aprosessipiste: vik:avirta L3
S1_J01_Q1		6003	0	104	S1_J01_Q1	vik:aprosessipiste: Kuormusvirta L1
S1_J01_Q1		6004	0	105	S1_J01_Q1	vik:aprosessipiste: Kuormusvirta L2
S1_J01_Q1		6005	0	106	S1_J01_Q1	vik:aprosessipiste: Kuormusvirta L3
S1_J01_Q1		6006	0	107	S1_J01_Q1	vik:aprosessipiste: vastaloelo
S1_J01_Q1		6007	0	108	S1_J01_Q1	vik:aprosessipiste: reaktanssilieto
S1_J01_Q1		6008	0	109	S1_J01_Q1	vik:aprosessipiste: resistanssilieto
S1_J01_Q1		6009	0	110	S1_J01_Q1	vik:aprosessipiste: vik:aluyppi
S1_J01_Q1A		10			S1_J01_Final	Final relocking

Identification:

Comment Text (CQ):

Object Text (DX, TX): Final relocking

Object Identifier (OI): S1\_J01\_Final

OPC Item Name (ON):

OPC Event Source (ES):

Operation State:

In Use (IU) Switch State (SS): 1 - Manual

Process Object Type:

Station/Object: EC:Double Indicat

Configurable: Dynamic All Attributes

Addresses | Alarm Generation | Alarm Handling | Post-Processing | **Events** | History | Printouts | Blocking | Misc

Event Channel:

Action Enabled (AE)

Action Name (AN): VIKA\_PAATTOTYO

Action Activation (AA): 1 - New Value

Action at First Update (AF)

Action on History Values (AH)

Event Object Enabled (EE)

Modification Time (ZT): 10-03-11 09:52:35

Flow: [Previous] [Next]

Buttons: OK Cancel Apply

**Kuva 30.** ”Lopullinen laukaisu” prosessipisteen sekä tapahtumakanavan asettelu.

Lopullinen laukaisu arvon vaihtuessa ykköseksi alkaa proseduurin ajo. Käynnistymisestä annetaan ilmoitus Notification ikkunaan.

```
#ERROR CONTINUE
#if %ov==1 #then #block
    @A=CONSOLE_OUTPUT(times + "vika_HAA:C proseduurin käynnistynyt")
```

**Kuva 31.** Proseduurin ajo prosessipisteen ”lopullinen laukaisu” object valuen arvon muuttuessa ykköseksi.

Seuraavaksi proseduuri kerää ala-asema kommunikaatioon tarvittavilta pisteiltä vikatiedot ja tekee niistä vektorit. Kuvassa 32 on esitettyinä vikavirtojen keruu pisteiltä. Muuttuja ”vv\_ln” on katkaisija looginen nimi, joka haetaan ”katk” muuttujalta. Näin saadaan aina oikea katkaisijan looginen nimi, jolta lopullinen laukaisu on tullut. Kuvassa pov6000 – pov6002 ovat prosessipisteet, joista luetaan arvot. Opinnäytetyön ohessa luodun toiminnallisuuden tekemät prosessipisteet ovat juuri kyseisiä indeksiltään 6000-alkuisia pisteitä.

```
;vikavirrat prosessipisteistä (FLOC lohkolta)
;=====
@vv_ln=%katk
@vv(1)='vv_ln'_'Q1:pov6000 ;vaiheen L1 vikavirta
@vv(2)='vv_ln'_'Q1:pov6001 ;vaiheen L2 vikavirta
@vv(3)='vv_ln'_'Q1:pov6002 ;vaiheen L3 vikavirta

@vikavirta=max(%vv(1), %vv(2),%vv(3))
@vv2=%vikavirta
```

**Kuva 32.** Tietojen haku prosessipisteiltä ja vektorin muodostaminen.

Kun kaikista arvoista on muodostettu vektorit, siirretään tiedot BDU\_FAULT-proseduurin muuttujiin. Tämän jälkeen ajetaan komennolla #DO BDU\_FAULT:C OPC- rajapintaa varten oleva proseduuri läpi kuvan 33 mukaisesti. /6/

```
@v_I_Msr=%vikavirta ;OIKOSULKUVIRTA
@e_Event_time='KATK'Q0:PRT10 ;TAPAHTUMA-AIKA
@i_Event_msr='katk'Q0:PRM10 ;TAPAHTUMAN MILLISEKUNNIT
@t_Source_ln='katk'+"Q0:PDB10" ;KATKAISIJA
@t_Phase=%vaihetieto ;vaihetieto
@t_Type=%vikatyyppi ;VIAN TYYPI
@t_Trigger="final" ;LOPULLINEN LAUKAISU
@t_Notation="real" ;TODELLINEN MITTAUSARVO
@r_I_Load=%kuormitusvirta ;KUORMITUSVIRTA
@v_method_ln="REF54X" ;REAKTANSSIPAIKANNUS
@v_RF_Msr=%vika_r ;RESISTANSSIARVO
@v_XF_Msr=%vika_x ;REAKTANSSIARVO

#DO BDU_FAULT:C ;VIKATIEDOSTO ORERALLE OIKOSULUSTA

@a=CONSOLE_OUTPUT("t_Source_ln 't_Source_ln'")
@b=CONSOLE_OUTPUT("t_Type 't_Type'")
@d=CONSOLE_OUTPUT("e_Event_time 'e_Event_time'")
@e=CONSOLE_OUTPUT("t_Trigger 't_Trigger'")
@f=CONSOLE_OUTPUT("t_Notation 't_Notation'")
@h=CONSOLE_OUTPUT("vikavirta 'vv2'")
@i=CONSOLE_OUTPUT("vaihetieto 't_Phase'")
@j=CONSOLE_OUTPUT("kuormitusvirta 'kv2'")
@k=CONSOLE_OUTPUT("Resistanssiarvo v_RF 'R'")
@l=CONSOLE_OUTPUT("Reaktanssiarvo v_XF 'X'")

#SET VIKAI:PB11=0 ;PAIKANNUS TEHTY
#block_end
```

**Kuva 33.** Arvojen tuonti ja BDU\_FAULT proseduurin ajo.

## **5.6 Onnistuneiden AJK ja PJK vikatietojen keräys**

Päättötyön loppuvaiheessa DMS 600 järjestelmää kehittävän ja projektoivan yksikön taholta esitettiin kehitysidea liittyen onnistuneiden PJK ja AJK tapahtumien vikatietojen tuomisesta DMS 600 järjestelmään. Nykyisissä järjestelmissä vikatiidot kerätään kun on tapahtunut lopullinen laukaisu, eli silloin kun vika on pysyvä. Tällaisissa tapauksissa ei voida toimintaa käsitellä ennaltaehkäisevänä, vaan vian paikannukseen ja korjaukseen liittyvänä toimenpiteenä.

Onnistuneiden PJK ja AJK kytkentöjen vikatietojen DMS 600 järjestelmään tuomisella saadaan toimintaa muutettua reagoivasta ennakoivaksi. Mikäli jollekin johto-osalle tulee paljon onnistuneita PJK tai AJK tapahtumia, voidaan päätellä johto-osalla olevan jokin ohimenevä vika, joka uusiutuu tietyin väliajoin. Tällainen vika voi olla tietyin aikavälein johtimeen osuva puun oksa, joka on esimerkiksi lumitaakan takia liikkunut huomattavan lähelle johdinta ja tietyin väliajoin johtoon osuessaan aiheuttaa hetkellisen maasulun. Tämä puolestaan aiheuttaa paljon onnistuneita PJK ja AJK tapahtumia. Tämän tyyppisten tapahtumien vikatietojen prosessoinnilla voidaan paikantaa vähintään lähtö, jolla kyseisiä vikoja ilmenee. Mikäli valitaan lähetettäväksi koko vikapaketti onnistuneesta PJK ja AJK tapahtumasta, voidaan päästä hyvin lähelle tarkkaa paikannusta. Reagoimalla edellä mainittuihin ohimeneviin vikoihin, voidaan ennaltaehkäistä suurempia vikoja, mikä puolestaan parantaa sähkönjakelun luotettavuutta.

### **5.6.1 Käytännön toteutus onnistuneille PJK ja AJK tiedoille**

Onnistuneiden PJK ja AJK tietojen keruu on hyvin samantapainen kuin lopullisen laukaisun yhteydessä suoritettava vikapaketin keruu ja siirtoketju. Releen AR5Func toimilohkosta on mahdollista saada ulos tieto onnistuneista jälleenkytkennöistä. Sekä AJK että PJK tapahtumille luodaan ala-asemakonfiguraatioon ristiviittaukset käytettävien protokollien mukaan. Ala-asemakonfiguraation kuvaukseen luodaan viittaukset kyseisille signaaleille SPA – IEC 60870-5-104 protokollien mukaisesti KU-2000 ala-asemalla.

Seuraavaksi luodaan prosessipisteet onnistuneita PJK ja AJK tietoja varten. Näiden pisteiden taakse luodaan omat tapahtumakanavat ja väliproseduurit. Väliproseduurit ovat samantyyppisiä kuin jo aiemmin esitelty lopullisen laukaisun ajama proseduuri. Työssä luotiin valmiit proseduurit onnistuneille PJK ja AJK kytkenöille.

Muutoksina lopullisen laukaisun siirtoproseduureihin ovat muiden vikatapahtumien blokkaukset pois. Lisäksi ”t\_Trigger” tieto on vaihdettu PJK tapahtumassa ”reclosing” tekstiksi ja AJK tapahtumassa ”delayed\_reclosing” tekstiksi. Lisäksi console output ikkunaan kirjoitettava teksti on muutettu kullekin proseduurille uniikiksi, jotta tiedetään mitä tapahtumaa kullakin hetkellä käsitellään.

S1_J01_Q1A	10			S1_J01	Q1lopullinenAJK	Final reclosing AJK
S1_J01_Q1B	10			S1_J01	Q1onnistunutAJK	Successful AJK
S1_J01_Q1P	10			S1_J01	Q1onnistunutPJK	Successful PJK

**Kuva 34.** Prosessipisteet, jotka on linkitetty tapahtumakanaviin ja proseduureihin.

Onnistuneiden PJK ja AJK vikatiетоjen tuominen valvomotasolle edistää vikojen ennaltaehkäisyä, mutta lisää myös informaatiotulvaa valvomo-operaattorille. Esimerkiksi rajun myrskyn aikana onnistuneiden PJK ja AJK tapahtumien määrä kasvaa rajusti, jolloin vastaavasti informaatio DMS 600-järjestelmässä kasvaa. Ongelmaan ratkaisuksi on tehty erilliset painikkeet luotuun konfigurointityökaluun, jonka avulla voidaan asettaa onnistuneiden PJK ja AJK prosessipisteiden tilatiedot nolliksi myrskyn ajaksi, jolloin kiireisimmän myrskyjakson aikana DMS järjestelmään tulee ainoastaan lopullisten laukaisujen tiedot. Edellä mainittu esto tapahtuu lisäämällä väliproseduureihin käyttöehtoihin object valuen lisäksi vaatimuksen, että estoprosessipisteen arvo tulee olla nolla, että proseduurin ajo sallitaan.

## 5.7 BDU\_FAULT proseduurin OPC rajapinnan käyttö

Toimiakseen oikein BDU\_FAULT proseduurin täytyy tehdä määrittäviä käytettävää rajapintaa varten. Rajapinnat ovat jo edellä mainitut OPC ja SCIL-API. Toinen näistä on pakko asettaa proseduurin. OPC rajapinta on uudempi ja helppo-

käyttöisempi vaihtoehto, jota tullaan tulevaisuudessa käyttämään. MicroSCADAn ja DMS 600-järjestelmien välillä on normaali server – client yhteys.

Yleensä tällaista server - client yhteyttä varten täytyy jokainen siirrettävä signaali luoda OPC pisteeksi, jolloin signaalin siirto onnistuu. Vikapakettien siirrossa on DMS 600 puolelle luotu sisäänrakennettu toiminto, joka lukee indeksiltään 5000 alkuiset prosessipisteet aina kun niiden tila päivittyy. Näin ollen signaaleja ei tarvitse luoda erikseen OPC pisteiksi, vaan sisäänrakennettu ominaisuus hoitaa pisteiden luennan DMS 600 järjestelmään.

OPC rajapintaa käytettäessä tulee huomioida BDU\_FAULT proseduurin valmius OPC sekä SCIL-API rajapinnalle. Proseduurissa on valmius molempien käyttöön. Proseduurin alkuosassa täytyy valita käyttöön toinen siirtotavoista. OPC rajapintaa varten täytyy proseduurin alkuosaan tehdä muuttujalle ”@t\_con\_type” kuvan 35 mukainen määrittely.

```

;Used communication type, SCIL-API or OPC (For compatibility existing SCIL-API selected as default)
@t_con_type="OPC" ;Define the type of connection: "SCIL-API" or "OPC"

```

**Kuva 35.** OPC rajapinnalle tehtävä määrittely.

Proseduurin keskiosassa on molemmille siirtotavoille valmiit komennot. Näistä toinen on joko poistettava tai jätettävä puolipisteillä kommenttiriveiksi.

## 5.8 Huomioitavaa proseduurien käytössä

Proseduurien avulla tehtävät vianpaikannustoimenpiteet ovat aina projektikohtaisia, johtuen prosessipisteiden loogisen nimen vaihtelevasta nimeämistavasta. Prosessipisteet ala-asemakommunikaatiota varten luodaan aina loogiselta nimeltään samoiksi kuin katkaisija kyseisellä lähdöllä. Ainoastaan pisteen OI attribuutti muutetaan 6000-alkuiseksi ja TX attribuuttiin kirjoitetaan viittaus vikaprosessipisteeseen. Väliproseduurin liipaisevan prosessipisteen tulee olla samaa muotoa kuin katkaisijoiden loogisen nimen.

Mikäli katkaisijoiden loogiset nimet ovat ”S1\_J01\_Q1”, täytyy liipaisevan pisteen olla muotoa ”S1\_J01\_Q1xx”. Tämä siksi, että väliproseduurin liipaisevan pisteen

loogisen nimen yhdeksän ensimmäistä merkkiä kirjoitetaan muistiin loogiseen nimeen komennolla @KATK = SUBSTR (%APU,1,9). Numero yksi tarkoittaa, että luku aloitetaan ensimmäisestä merkistä ja luku lopetetaan yhdeksänten merkkiin, jolloin muistiin otetaan "S1\_J01\_Q1". Lainausmerkkien sisään tulee yhdeksän merkkiä ja xx jää pois, jolloin muistiin otettu looginen nimi on sama kuin indeksiltään 6000-alkuisilla pisteillä. Väliproseduurissa käsiteltävä tieto otetaan esimerkiksi komennolla; "@vv(1)='vv\_ln':pov6000 ;Vaiheen L1 vikavirta", jolloin "vv\_ln" on liipaisevan pisteen yhdeksän ensimmäistä merkkiä ja perään kirjoitetaan pov6000, jolloin saadaan haettua tieto ala-asema pisteeltä. Mikäli loogiset nimet poikkeavat esitetyistä, täytyy kerättävien merkkien pituutta muuttaa SUBSTR-komentoon.

Vikatietojen yhteyteen kirjoitetaan yhdeksi muuttujaksi "method", jonka tulee olla komentoproseduurissa määritelty samaksi kuin vastaava muuttuja DMS 600 järjestelmän puolella. Tämä metodin nimi voi olla mikä tahansa kirjain- ja numero-yhdistelmä.

## 6 VIANPAIKANNUSDIALOGI

Työn aikana luotujen lukuisten proseduurien käyttö muodostuu käytännön tasolla erittäin haastavaksi. Proseduurit täytyy ajaa tietyssä järjestyksessä ja kullekin eri rele- ja protokollatyypille täytyy luoda erillisiä muutoksia proseduureihin.

Ratkaisuksi sekavaan tilanteeseen luotiin työn ohessa erillinen Visual SCIL ohjelmoinnilla toteutettu konfigurointityökalu MicroSCADAn työkaluvalikkoon, joka muutamalla yksinkertaisella valinnalla suorittaa proseduurit oikeassa järjestyksessä automaattisesti. Työkalun käyttöliittymä on koodattu käyttäen Visual SCIL työkalua. Pääohjelmassa tehtyjen valintojen perusteella ohjelma osaa itse ajaa oikeat proseduurit tapauskohtaisesti. Luvussa viisi esitetyt, sekä erikseen luodut proseduurit toimivat hallitusti konfigurointityökalussa tehtyjen valintojen perusteella.

### 6.1 Visual SCIL

Visual SCIL on normaalia SCIL-ohjelmointikieltä muistuttava ohjelmointikieli, jonka avulla pystytään luomaan ikkunoita, taulukoita, nappeja ym. visuaalisia objekteja, joiden taakse saadaan luotua toiminnallisuutta. Selkein ero normaaliin SCIL-ohjelmointiin on käskyjen anto piste-alkuisesti. Lisäksi eri objektien arvoja voidaan käsitellä omilla käskyillä, jotka ovat niin ikään piste-alkuisia. Visual SCIL toimii taso-tyyppisesti, joka tarkoittaa, että toimintoja voidaan luoda eri tasoille ja siirtää tarpeen mukaan alemmille tasoille .root tai .parent toiminnoilla. Visual SCILin main program ikkunan ”method” välilehteen voidaan luoda eri nimiä metodeja, joiden sisään voidaan rakentaa SCIL toiminnallisuuksia. //

### 6.2 Dialogin kuvaus

Dialogin valintaikkuna on suunniteltu mahdollisimman yksiselitteiseksi. Dialogin yläosassa suoritetaan valinnat ala-asemalle tehtäviä prosessipisteitä varten. Tässä valinnassa otetaan kantaa FLOC- ja SCEFRFLO-lohkoihin, releen tyyppiin ja käytettävään protokollaan. Työkalu on rajoittunut releiden ja protokollien osalta, mutta tukee käytetyimmät valvomotason protokollat sekä suositut REF 543- ja

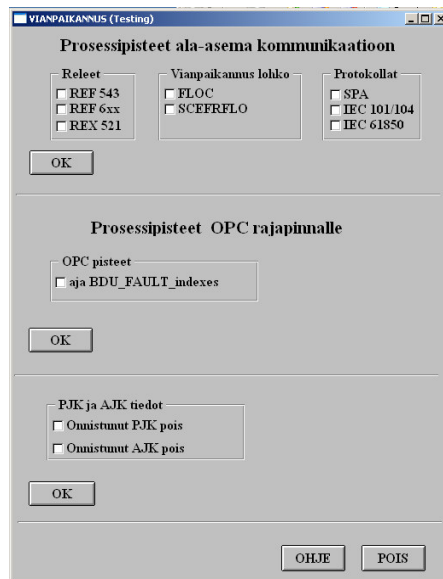
REF630 releet. Työkaluun on sisäänrakennettu toiminnot, jotka estävät tekemästä väärän tyyppisiä valintoja, kuten valitsemalla FLOC lohkon ja releen, jossa ei ole kyseistä lohkoa.

Mikäli valinnaksi suoritetaan REX 521, luodaan vain vikavirralle prosessipisteet, koska kyseinen suojarole ei sisällä FLOC-lohkoa. Ratkaisu on hieman kyseenalainen johtuen suojauslohkon tyylistä ottaa vikavirrat ylös syklisesti kolmeen eri rekisteriin. REX 521 valinta työkalussa edellyttää, että relekonfiguraatioon on luotu ajastin jokaisen suojaustoiminnon muistirekisterille, joka puolestaan blokkaa kolmipaikkaisen muistirekisterin yksipaikkaiseksi. Tällöin saadaan poimittua viimeisin vikavirran arvo. Toinen vaihtoehto on vertailla eri muistirekisterien aikaleimoja, jolloin saadaan poimittua lähimpänä lopullista laukaisua aikaleimaltaan olevat tiedot. REF 543- ja 600-sarjan releiden FLOC- ja SCEFRFLO-lohkoissa on eroja, mutta työkalu luo 500-sarjan FLOC-lohkon tarpeisiin pisteet, koska DMS-järjestelmä hyödyntää ainoastaan tämän lohkon tietoja.

Dialogin keskiosassa on erillinen alue, jossa valitaan BDU\_FAULT\_indexes ajo. Näin saadaan samasta ikkunasta suoritettua OPC- rajapinnalle kaikki prosessipisteet yhdellä napin painalluksella.

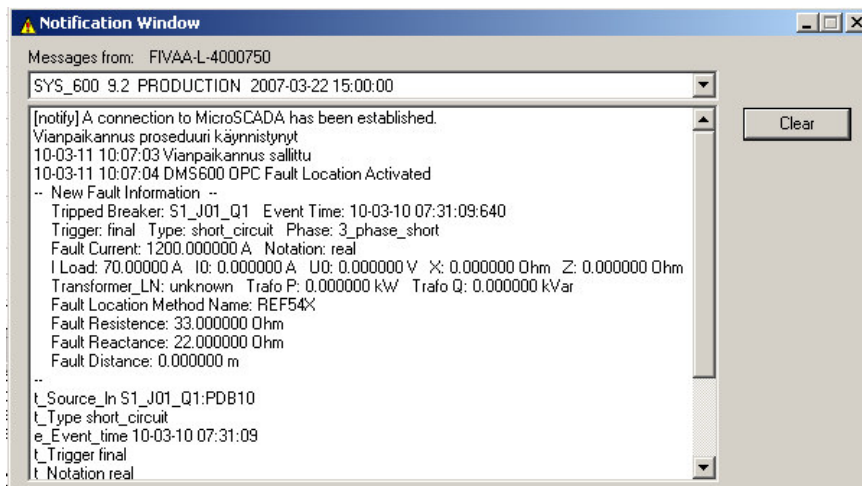
Työkalussa on myös toiminto, jonka avulla voidaan ottaa pois käytöstä onnistuneiden PJK ja AJK signaalien tuonti valvomoon. Kyseisen toiminnon käyttö edellyttää kahden erillisen prosessipisteen luontia ja niiden nimeämistä btn\_ok\_blokkaus metodin taakse.

Ohje painikkeella saadaan avattua erillinen ikkuna, jossa on annettuna tarkat ohjeet työkalun käyttöön. Lisäksi pois painikkeella saadaan suljettua työkalu.



**Kuva 36.** Dialogin valintaikkuna.

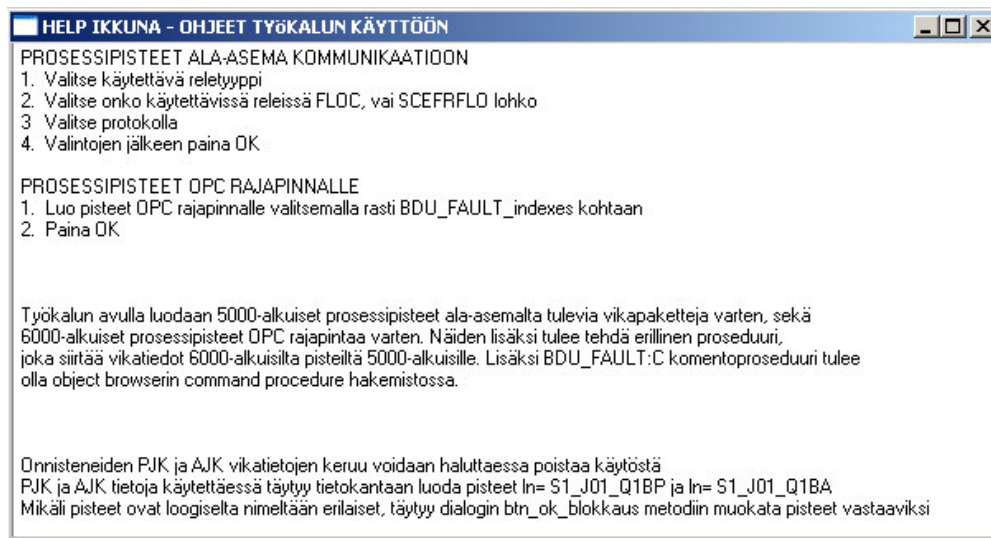
Kaikki suoritettavat valinnat, sekä toiminnot annetaan console\_outputin kautta Notification-ikkunalle, jonka ansiosta työkalua käytettäessä voidaan tarkastella mitä kaikkea työkalu tekee kussakin vaiheessa. Kuvassa 37 on esitettyä Notification-ikkunan informaatio, kun koko prosessi on onnistunut reletasolta aina OPC rajapinnalle asti. Työkalua käytettäessä Notification-ikkunaan kommentoidaan myös luodut pisteet.



**Kuva 37.** BDU\_FAULT ajettu komentoproseduurilla lopullisen laukaisun ohjaamana.

Dialogin suorittavat komennot voidaan jakaa kahteen osaan. Molemmat komentoketjut toimivat OK painikkeiden takana. Kun suoritettujen valintojen jälkeen painetaan OK painiketta, tarkistetaan kaikkien valintojen tilatiedot ja niiden eri yhdistelmät. Edellä mainittujen arvojen perusteella ajetaan oikeat proseduurit ta-pauskohtaisesti. Lisäksi eri painikkeiden takana on ehtoja, jotka vaikuttavat tulevien valintojen mahdollisuuksiin. Esimerkiksi valittaessa FLOC-lohko, muuttuu REX521 valinta disable tilaan koska kyseinen rele ei sisällä FLOC-lohkoa. Samoin valittaessa jokin rele, muuttuvat muut relevalinnat disable tilaan. Näin saadaan estettyä ohjelmaa käyttävää henkilöä tekemästä vääriä valintayhdistelmiä.

Ohje painikkeen avulla aukeaa erilliseen ikkunaan ohjeet työkalun käyttöön sekä selitykset, mitä prosessipisteitä työkalu luo. Help-ikkuna sulkeutuu rastilla oikeasta yläkulmasta. Kuvassa 38 on esitettynä Help-ikkuna.



**Kuva 38.** Ohje painikkeesta avautuva Help-ikkuna.

## 7 TULEVAISUUDEN HAASTEET

Vianpaikannusprosessissa MicroSCADAssa tehtävät toimenpiteet ovat haastavia, mutta täysin mahdollisia. Proseduurien hallittu käyttö vaatii kuitenkin tietämystä MicroSCADA järjestelmästä. Vianpaikannuksessa käytönvalvontajärjestelmä ei muodosta pullonkaulaa, vaan päinvastoin lisää mahdollisuuksia. Esimerkiksi onnistuneiden PJK ja AJK tietojen käyttö vaatii erillistä älyä, joka on helppoa luoda MicroSCADAn proseduureihin. Toisaalta vanhoja järjestelmiä päivitettäessä lukuisat eri proseduurit luovat haasteita. Tulevaisuudessa tämän tyyppisten ominaisuuksien rakentaminen suojaan FLOC- ja SCEFRFLO-lohkoon olisi yksi keino selkiyttää vianpaikannusprosessia, jolloin MicroSCADAan tehtävä äly vähenisi ja rooli datan välittäjänä paranisi. Työssä luodun dialogin avulla prosessi selkiytyy konfiguraatiota tekevän henkilön näkökulmasta, mutta vastaavasti lisää vikatiedon prosessointia huomattavasti käytönvalvontajärjestelmässä. Tulevaisuudessa on tärkeää valita, painotetaanko tiedon prosessointia rele, vai käytönvalvontajärjestelmän tasolle.

Rajoittavat tekijät ovat releiden kyky tehdä vikoihin liittyvää laskentaa sekä käytöntukijärjestelmän kyky prosessoida tietoa sadoilta tai jopa tuhansilta releiltä. Releiden osalta Fault Location-toimilohkot ja niiden kehittäminen ovat tärkeässä asemassa vianpaikannuksen parantamisessa.

Suurin haaste on maasulun paikannuksen tarkkuus. Maasulun paikannuksen liittyvä kehitystyö on tulevaisuudessa sekä relepuolella että käytöntukijärjestelmässä suuressa roolissa. Nykyinen maasulun paikannus ei ole aina riittävän tarkka kaikissa tilanteissa johtuen maan resistiivisyyden vaihteluista. Pelkästään 10:n prosentin heitto maasulun reaktanssiarvossa aiheuttaa keskimäärin 20 km vääristymän paikannuksessa. /10/ Lisäksi verkostolaskennassa saatavat arvot ovat osittain viitteellisiä johtuen samasta ongelmasta. Ongelman korjaamiseksi reletasolla testataan uusia laskenta-algoritmeja, joilla paikannusta voidaan parantaa. Toisaalta nykyisin releen FLOC- ja SCEFRFLO-lohko antavat paikannuksen reaktanssi ja vikavirta-arvolle tarkkuustiedon, jolloin halutessa voidaan suhtautua saatuun paikannukseen kriittisesti.

## 8 JOHTOPÄÄTÖKSET

Työssä olevat ratkaisut on tehty tulevaisuuden tarpeita ajatellen. Tutkitut releet, protokollat sekä tiedonsiirtotavat tukevat suurelta osin tulevaisuudessa käytettäviä ratkaisuja. Luotuja työkaluja voidaan halutessa laajentaa myös muille protokollille, mikäli tarvetta esiintyy. Työ sisältää vikapakettien keruuseen ja siirtoon vaikuttavat osatekijät. Lisäksi luotu konfigurointityökalu selkiyttää ja helpottaa MicroSCADA:ssa tehtäviä toimenpiteitä sekä proseduurien käyttöä.

Onnistuneiden PJK ja AJK vikatietojen tuominen valvomotasolle edistää vikojen ennaltaehkäisyä. Lisäksi poiskytkentämahdollisuus parantaa myrskyn aikaisen informaation hallintaa. Kyseisen ominaisuuden käytöstä tulee neuvotella tarkemmin asiakaskohtaisesti.

Vianpaikannukseen liittyy karkeasti jaoteltuna kolme eri osatekijää: releessä tapahtuva vikatietojen prosessointi, käytönvalvontajärjestelmässä tapahtuva keruu ja siirto eteenpäin sekä käytöntukijärjestelmässä tapahtuva tiedon prosessointi. Kaikki kolme vianpaikannukseen vaikuttavaa osatekijää toteutetaan eri yksiköiden alaisuudessa. Reletasolla tehtyjen vianpaikannustoimintojen hyödyntäminen käytöntukijärjestelmässä on riippuvainen käytönvalvontajärjestelmän kyvystä kerätä ja lähettää tietoja eteenpäin sekä käytöntukijärjestelmän kyvystä hyödyntää tietoja. Työssä kerätyn tiedon perusteella suosittelen vianpaikannukseen liittyvän kehityksen olevan enemmän sidonnaista käytöntukijärjestelmän tarpeisiin. Mikäli releisiin kehitellään uusia vianpaikannukseen liittyviä menetelmiä ovat ne turhia, ellei käytöntukijärjestelmä pysty niitä hyödyntämään. Eri yksiköissä tapahtuvan kehitystyön tavoitteiden yhtenäisyys luo edellytykset tehokkaaseen vianpaikannusprosessin kehittämiseen.

## LÄHTEET

- /1/ ABB manuaalit, DMS 600 integration with SYS 600
- /2/ ABB manuaalit, Fault locator, 1MRS755454
- /3/ ABB manuaalit, REF 543 Technical manual
- /4/ ABB manuaalit, REF630IEC Technical manual
- /5/ ABB manuaalit, SPA osoitteistot, REF541\_543ParlistEN
- /6/ ABB manuaalit, SYS600\_SCIL Programming Interface
- /7/ ABB manuaalit, SYS600\_Visual Scil Application design
- /8/ ABB manuaalit, SYS600\_Visual Scil Objects
- /9/ ABB manuaalit, Tekninen ohje vakiokonfiguraatit, REX 521, 1MRS, 75508, viitattu 2.3.2009
- /10/ Ahola, Jussi, projektipäällikkö 19.2.2010, ABB Oy, Tampere. Haastattelu. DMS 600-järjestelmän tarpeet ja kehitysehdotukset
- /11/ Blackburn, J. Lewis & Domin, Thomas J 2007. Protective Relaying, Principles and Applications. CRC PRESS
- /12/ Energiamarkkinavirasto, Ohjeistus 19.2.2007 [online]. [viitattu 15.3.2010]. Saatavilla [www-muodosa: <URL:http://www.energia.fi/fi/tilastot/keskeytystilastot/keskeytystilasto-ohjev3.1.pdf>](http://www.energia.fi/fi/tilastot/keskeytystilastot/keskeytystilasto-ohjev3.1.pdf).
- /13/ Tyni, Jukka-Pekka, projektipäällikkö 10.3.2010, ABB Oy, Vaasa. Haastattelu. COM 600 ristiviittaukset
- /14/ Lidman, Vesa, projektipäällikkö 9.3.2010, ABB Oy, Vaasa. Haastattelu. PJK tietojen käsittely

- /15/ Mattila, Jouko, Kuumic KU2000 Ohjelmointi, 2001
- /16/ Mäkinen, Olavi, 2008, Opetusmateriaali, Sähköverkot luento-  
monisteet