



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Sami Selkänaho

OPENIEC61850 -KIRJASTON KÄYTTÖÖNOTTO

Tekniikka ja liikenne
2015

TIIVISTELMÄ

Tekijä	Sami Selkänaho
Opinnäytetyön nimi	OpenIEC61850 -kirjaston käyttöönotto
Vuosi	2015
Kieli	suomi
Sivumäärä	68
Ohjaaja	Jukka Matila

Tämän opinnäytetyön tarkoituksena oli tehdä työ, jossa tutkitaan IEC 61850 -standardia sekä siihen perustuvan OpenIEC61850 -kirjaston käyttöä. Työ on tehty Vaasan ammattikorkeakoululle. Toissijaisena tehtävänkuvana on ollut, että koulu voisi käyttää työn osia opetusmateriaalina koulutuksessa.

IEC 61850 on sähköautomaatiossa käytettävä tiedonsiirtostandardi. Se tarjoaa standardisoidun tavan ilmaista koko sähköaseman toiminnallisuutta. Tämän ansiosta eri valmistajien laitteet voivat kommunikoida keskenään. OpenIEC61850 -kirjasto on avoimen lähdekoodin Java -kielinen toteutus tästä standardista. Kirjastolla voidaan tuottaa ohjelmia, jotka käsittelevät IEC 61850 -palvelinten sisäistä dataa.

Tuloksena luotiin sovellus, joka hakee, kirjoittaa ja visualisoi dataa laitteesta tai SCL -tiedostosta. Työstä myös tehtiin wikisivusto, jossa käydään läpi tiivistetysti standardin ja kirjaston olennaisimpia osia.

ABSTRACT

Author	Sami Selkäinen
Title	Implementation of OpenIEC61850 Library
Year	2015
Language	Finnish
Pages	68
Name of Supervisor	Jukka Matila

In this thesis the task was to investigate the IEC 61850 standard and OpenIEC61850 library which is based on the standard. The work was made for Vaasa University of Applied Sciences. The secondary task was to develop learning material for the university of applied sciences.

IEC 61850 is a communication standard for electrical substations. It offers a standardized way to express the functionality of a complete substation. Thanks to this the devices of different manufactures may communicate with one another. OpenIEC61850 is an open source Java library based on the standard. With the library you can develop software that is able to process the internal data of IEC 61850 servers.

As a result a software was created that fetches, writes and visualizes the data of an Intelligent Electronic Device (IED) or an SCL file. A webpage which covers the crucial points of the standard and library was also made of the work.

Keywords	IEC 61850, OpenIEC61850, electrical substation, protective relaying
----------	---

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

TERMIT JA LYHENTEET

1	JOHDANTO.....	11
2	TAUSTA.....	12
	2.1 Suojareleiden kommunikaatio.....	12
	2.2 Käytetyt laitteet.....	13
3	IEC 61850.....	14
	3.1 Yleistä.....	15
	3.2 Datamalli.....	17
	3.2.1 Loogiset nodet.....	19
	3.2.2 Objektin nimeäminen.....	24
	3.2.3 Yleiset dataluokat.....	24
	3.3 ACSI.....	26
	3.3.1 Functional constraint.....	30
	3.3.2 CST.....	32
	3.4 SCSM.....	36
	3.4.1. GOOSE.....	37
	3.4.2 MMS.....	38
	3.4.3 SV.....	38
	3.5 SCL.....	39
	3.5.1 SCL -työkalut.....	41
4	KIRJASTOT.....	45
	4.1 Yleistä.....	45
	4.2 OpenIEC61850.....	45
	4.3 libiec61850.....	47
5	SOVELLUS.....	49
	5.1 Yleistä.....	49
	5.2 Toiminta.....	51
6	TULOKSET.....	60
	6.1 Datan nouto palvelimelta.....	60

6.2 Datan nouto SCL -tiedostosta.....	63
6.3 Write -funktio.....	65
7 JOHTOPÄÄTÖKSET.....	66
LÄHTEET.....	67

KUVALUETTELO

Kuva 1. Sähkönjakeluverkon hierarkia	s.12
Kuva 2. Standardin olennaisimmat osat	s.15
Kuva 3. IEC 61850:n rakenne	s.17
Kuva 4. Datamallin eri tasot	s.18
Kuva 5. Standardin datamalli	s.23
Kuva 6. Objektin nimen rakenne	s.24
Kuva 7. Datamallin luokkakuva	s.25
Kuva 8. ACSI:n malliluokat ja niiden palvelut	s.29
Kuva 9. Palveluiden seuraamiseen käytetty CST -luokka	s.32
Kuva 10. SCL -tiedoston data	s.41
Kuva 11. Single line diagram (SLD)	s.42
Kuva 12. SCL -tiedoston konfiguraatioprosessi	s.43
Kuva 13. OpenIEC61850 -kirjaston komponentit	s.46
Kuva 14. clientgui -projektin sisältö	s.50
Kuva 15. ClientSap -instanssin luonti	s.52
Kuva 16. Serverin datamallin noutaminen käyttämällä assosiaatiota	s.53
Kuva 17. Datamallin nouto SCL -tiedostosta	s.54
Kuva 18. serverModelin jäsentäminen	s.55
Kuva19. addLogicalNode() -metodi	s.55
Kuva 20. Ohjelman graafinen käyttöliittymä	s.56

Kuva 21. Luokkakaavio ohjelman toimintaperiaatteesta	s.58
Kuva 22. Ohjelman konsoli	s.59
Kuva 23. Vamp 257- suojarahleen SCL- tiedosto	s.60
Kuva 24. Vamp 257 -releen data ja tunnistuskilven kuvaustieto	s.61
Kuva 25. Vamp 257 -releen data ja SWI -noden Pos -data	s.62
Kuva 26. LLN0.DSG1 -datasetin data	s.63
Kuva 27. Tiedoston avausdialogi	s.64
Kuva 28. Example.icd:n data	s.64
Kuva 29. Write -funktion käyttö SampleServerillä	s.65
Kuva 30. Kirjoituspyynnön vastaanottaminen	s.65

TAULUKKOLUETTELO

Taulukko 1. Loogisten nodejen ryhmiä	s.19
Taulukko 2. Loogisten nodejen ryhmiä	s.22
Taulukko 3. Functional constraintit	s.30
Taulukko 4. SCL:n tiedostotyytit	s.39

TERMIT JA LYHENTEET

ACSI	Abstract Communication Service Interface, IEC 61850:n palvelurajapinta
Assosiaatio	Clientin ja serverin välille luotu kuljetusväylä
Attribuutti	Nimetty, tietyn tyyppinen dataelementti
CDC	Common Data Class, IEC 61850:n osa, joka luettelee nk. yleiset dataluokat
Data	Rakenteellista, IED:ssä sijaitsevaa tietoa, jota voidaan lukea tai kirjoittaa
Data-attribuutti	Määrittää datan nimen, formaatin, mahdolliset arvot ja arvojen esittämisen kommunikaatiossa
Dataobjekti	Loogisen noden osa, joka edustaa tiettyä informaatiota, kuten tilaa tai mittausarvoa
Dataluokka	Luokka, joka kokoaa yhteen dataa ja data-attribuutteja
FC	Functional constraint, data-attribuutin ominaisuus, joka määrittää mitä palveluita data-attribuuttiin voidaan soveltaa
FCDA	Functional constraint data, referenssi datakokoelmaan, joilla on kaikilla sama functional constraint
FCDA	Functional constraint data attribute, referenssi yksittäiseen data-attribuuttiin, johon tietty functional constraint pätee

Funktio	Aseman suorittama tehtävä, joka yleensä vaihtaa dataa muiden funktioiden kanssa
GOOSE	Generic Object Oriented Substation Event, IEC 61850:n käyttämä protokolla, jolla välitetään tietoa aseman tapahtumista horisontaalisesti Ethernetin kautta
HMI	Human Machine Interface, näyttöruutu joka esittää aseman dataa loogisessa muodossa, jota käyttäjä voi käsitellä
IED	Intelligent Electronic Device, mikroprosessoripohjainen laite sähköasemalla, kuten suojarele
Jaettu funktio	Funktio, jonka kaksi tai useampi looginen node, jotka sijaitsevat eri fyysisissä laitteissa, suorittavat yhdessä
LAN	Local Area Network, tietyn pienen alueen, kuten rakennuksen tai teollisuusalueen kattava kommunikaatioverkko
Log	Tallenne kronologisesti järjesteltyä dataa
Looginen laite -luokka	Virtuaalinen laite, jonka tarkoitus on mahdollistaa toisiinsa liittyvien loogisten nodejen kerääminen yhteen, sekä tehdä hyödyllisiä listoja usein käytetystä tai viitatusta tiedosta
Looginen node	Pienin osa funktiota, joka välittää dataa
Palvelu	Jokin toiminnallisuus, kuten lukeminen tai kirjoittaminen, jolla käsitellään ja hallitaan dataa

Parametrit	Muuttujia, jotka määrittävät sähköaseman funktioiden ja IED:ien käyttäytymisen
Protokolla	Erä sääntöjä, jotka määrittävät toiminnallisten osien käyttäytymisen kommunikaation saavuttamiseksi
SLD	Single Line Diagram, yksinkertaistettu kuvio sähköjärjestelmästä
XML	Extensible Mark-up Language, rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja datakokonaisuuksia

1 JOHDANTO

Sähkönjakeluverkossa ala-aseman tarkoitus on vastaanottaa virtaa useilta linjoilta, muuntaa se alempijännitteiseksi ja jakaa se edelleen kuluttajille. Suojareleet tarkkailevat sähkövirtaa ja suojelevat verkkoa vikatilanteessa vahingoilta. Yleisin toiminto on virran katkaisu. Nykymaailmassa sähköverkot ovat kasvaneet suuriksi ja alalla toimii useita eri sähköntarjoajia. Eri yhtiöillä voi olla käytössä erilaisia laitteita ja perinteisenä ongelmana on ollut, että eri jakelijoiden laitteisto ei ole yhteensopivaa eikä kykene kommunikoimaan keskenään.

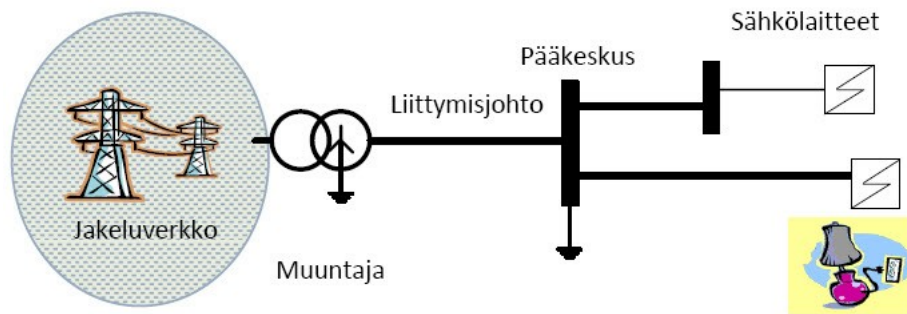
IEC 61850 -standardi pyrkii vastaamaan tähän. Se luo koko sähköasemalle yhteisen kielen, jonka avulla laitteet voivat kommunikoida keskenään. Tehdessään tämän se kirjoittaa yksien kansien sisään koko sähköaseman toiminnan.

Sähköasemaan kuuluu monenlaista toimintoa aina mittaustietojen lukemisesta kytkimen sulkemiseen. Näitä toimintoja voidaan suorittaa standardia hyödyntävien ohjelmistojen kautta. Monilla suuremmilla firmoilla on tätä varten itsekehityt kirjastot, mutta myös avoimen lähdekoodin kirjastoja on olemassa useammalle eri ohjelmointikielelle. Tässä työssä käytetty OpenIEC61850 on Java-kielinen avoimen lähdekoodin IEC 61850 -kirjasto, joka on lisensoitu GNU:n kevyemmällä yleislisenssillä.

2 TAUSTA

2.1 Suojareleiden kommunikaatio

Sähköasemalla ja niiden releillä on hierarkia, joka määräytyy sen mukaan, missä kohtaa verkkoa ne sijaitsevat. Lähempänä sähköntuottolähdettä oleva asema on hierarkisesti ylempänä kuin kauempana. Mitä kauemmaksi lähteestä mennään, sitä enemmän asemat haarautuvat suureksi verkoksi. Hierarkisesti alimpana ovat täten kuluttajaa lähimpänä olevat asemat. Eri asemat kommunikoivat ja välittävät tietoa keskenään. Ne välittävät tietoja vioista sähköjakoverkossa toisilleen ja voivat käskä esimerkiksi laittamaan virran poikki jos on tarpeen. Kuvassa 1 on kuvattuna tämä sähköjakoverkko hierarkia.



Kuva 1. Sähköjakoverkko hierarkia. /3/

Samantason tai samankaltaisen tason releiden välistä kommunikaatiota kutsutaan horisontaaliseksi kommunikaatioksi. Käytännössä tämä tarkoittaa samalla asemalla sijaitsevien laitteiden välistä kommunikaatiota. Hierarkisesti eri tasoilla olevien releiden kommunikaatiota kutsutaan vertikaaliseksi kommunikaatioksi. Kun suojarele kommunikoit releiden kanssa, jotka ovat sen ala- tai yliasemilla, kommunikaatio on vertikaalista. Jotta eri jakelijoiden IED:t toimisivat keskenään, niiden pitää pystyä kommunikoimaan keskenään. Tämä on ollut perinteisesti vaikeaa, koska eri jakelijoiden käyttämät laitteet käyttävät erilaisia

kommunikaatioprotokollia.

2.2 Käytetyt laitteet ja konfiguraatio

Tässä työssä on käytetty VAMP 257 -suojarelettä. Siinä on 28 digitaalituloa ja 13 digitaalilähtöä ja se sopii kaikkiin sähköjakeluverkon kohteisiin. Tässä työssä on keskitytty sen IEC 61850 -ominaisuuksiin.

Työtä varten laitetta on jouduttu konfiguroimaan vain hieman, jotta tiedonsiirto olisi mahdollinen. Laitteen Bus -valikon alalehdellä Ethernet Port voidaan konfiguroida laitteen IP ja portti. IP tulee asettaa niin, että se on verkkokortin kanssa samassa aliverkossa, jotta ne voivat kommunikoida. Tämän yksinkertaisen vaiheen jälkeen laite on ollut valmis IEC 61850 -kommunikaatiota ja täten tätä työtä varten. Työssä on myös käytetty ohjelmistoa, joka imitoi IEC 61850 -palvelinta sekä SCL -tiedostoja.

3 IEC 61850

3.1 Yleistä

IEC 61850 on sähköasemien suunnittelussa käytetty standardi. Siitä tuli kansainvälinen standardi vuonna 2004. Kun standardia lähdettiin tekemään, tavoitteeksi asetettiin, että standardi mahdollistaisi eri jakelijoiden IED:ien kommunikaation keskenään siten, että olisi mahdollista suorittaa erilaisia toimintoja niiden välillä, kuten ylivirtasuojauksia. Toisin sanoen siis - laitteet olisivat yhteensopivat. Standardia varten luotu projektiryhmä työskenteli vuodesta 1995 lähtien sen parissa. Kaiken kaikkiaan tavoitteiksi, jotka myös saavutettiin, asetettiin seuraavat kohdat:

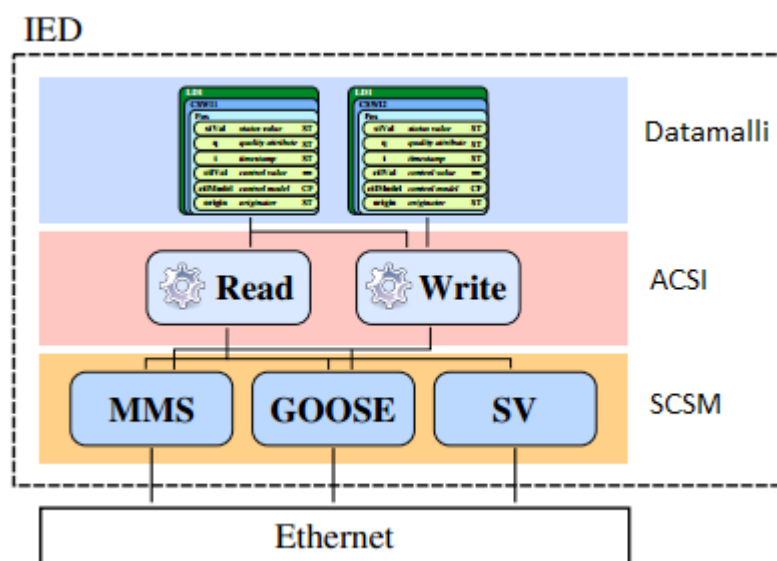
- yksi protokolla koko sähköasemalle
- peruspalveluiden määrittäminen datan siirrolle
- suuremman yhteensopivuuden saavuttaminen eri sähköntarjoajien välille
- yhteinen metodi datan tallentamiseen
- testauksen täydellinen määrittäminen yhteensopiville laitteille. /4/

IEC 61850:n metodi ratkaista eri jakelijoiden välisen kommunikaation ongelma perustuu kolmeen pääkomponenttiin, jotka on kuvattuna kuvassa 2. Ensimmäinen on sen datamalli, joka kuvaa relesuojauksen kannalta olennaisia tietoja ja funktioita laitteessa ja luo laitteille yhteisen ”kielen”. Tällä hetkellä IEC 61850 on ainoa standardi, joka käyttää standardisoitua datamallia.

Toinen komponentti on ACSI (Abstract Communication Service Interface), johon kuuluvat standardin toiminnot ja operaatiot. Näitä toimintoja kutsutaan palveluiksi. Palvelujen avulla käsitellään sähköaseman dataa.

Kolmas komponentti on SCSM (Specific Communication Service Mapping), joka määrittää kuinka datamalli ja ACSI -komennot enkoodataan, eli minkälaista kommunikaatioprotokollaa käytetään. Protokollia, joita tällä hetkellä käytetään on

kolme: MMS, GOOSE, ja SV. MMS:ää käytetään hierarkisesti eritasoisten laitteiden kommunikaatiossa (vertikaalinen kommunikaatio) välittämään reaaliaikaista dataa. MMS:ää käytetään enemmän silloin, kun on tarkoitus tuoda sähköaseman dataa ihmiselle luettavaksi ja käsiteltäväksi johonkin rajapintaan. GOOSEa käytetään sähköaseman tapahtumien välittämisessä samantasoisten laitteiden välillä. Se automatisoidaan sähköasemalle. SV:ä käytetään analogisten mittausarvojen välittämisessä näytteistettynä samantasoisten laitteiden välillä asemalla.



Kuva 2. Standardin olennaisimmat osat. /18/

Laitteiden keskinäisen toiminnallisuuden ja muiden edellämainittujen tavoitteiden lisäksi standardin tarkoituksena on tarjota vapaa konfiguraatio. Tämä tarkoittaa sitä, että laitteita voidaan konfiguroida mahdollisimman vapaasti ja että se toimii yhtä hyvin keskitetyille ja epäkeskitetyille järjestelmille. Konfiguraatioon IEC 61850:ssä käytetään sen omaa XML-pohjaista kieltä SCL:ää (Substation Configuration Language).

Yhteensopivuuden ja vapaan konfiguraation lisäksi vielä kolmantena tärkeänä osana IEC 61850:n tarkoituksia on sen tavoite pitkäjänteisyyteen. Tarkoituksena on, että standardi kykenee seuraamaan kommunikaatioteknologian kehityksen perässä ja vastaamaan kasvaviin järjestelmävaatimuksiin. Sähköasemille on

olemassa maailmanlaajuisesti kymmeniä eri standardeja, mutta IEC 61850 on ainoa, joka tarjoaa standardisoidun tavan usean tarjoajan sähköverkkojen laitteiden väliseen kommunikaatioon, joten sillä on täten melko ylivoimainen asema tulevaisuudennäkymiä ajatellen.

IEC 61850 on kokonaisuudessaan määritetty kymmeneen eri osaan, jotka muodostavat yhteensä 15 dokumenttia. Osat 6-10 ovat standardin pääosat.

Ensimmäinen osa, IEC61850-1, on johdatus, jossa käydään läpi standardin peruseriaatteita, annetaan yleiskatsaus standardiin ja kerrotaan kuinka standardin tekoa on lähdetty lähestymään. Siinä puhutaan standardin taustasta, kuten todetaan laitteiden yhteensopivuuden vaatimus nykyaikaisissa sähköasemissa ja käydään läpi standardin kehityksen historiaa. Perusteoria, kuten loogiset nodet, funktiot, ACSI ja CDC (Common Data Class) esitellään.

IEC61850-2:ssa selitetään suurin osa standardiin liittyvien termien tarkoituksista. Olennaisimmat siinä esiintyvät termit on selitetty tämän työn alussa.

IEC61850-3:ssa käsitellään sähköasemien kommunikaatioverkkoja ja -järjestelmiä. Siinä määritellään laitteiden välinen kommunikaatio ja siihen liittyvät vaatimukset. Tällaisiin vaatimuksiin kuuluu esimerkiksi, että kommunikaatiossa tapahtuvat virheet eivät vaaranna suoja-funktioita. Osassa painotetaan laatuvaatimuksia ja esimerkiksi ympäristölliset asiat on selvitetty. /6/

IEC61850-4:ssä käsitellään sitä, kuinka järjestelmää tai projektia ylläpidetään IEC 61850:lla. Siinä määritellään järjestelmän teknisiä vaatimuksia. Esimerkiksi järjestelmän IED:t ja niiden suhteet toisiinsa tulee kartoittaa. Käytännössä tämä toteutetaan SCL:llä. SCL:ään liittyy myös erilaisia työkaluja, esimerkiksi dokumentaatiotyökalu, ja osa näistä on käsitelty tässä osassa. /7/

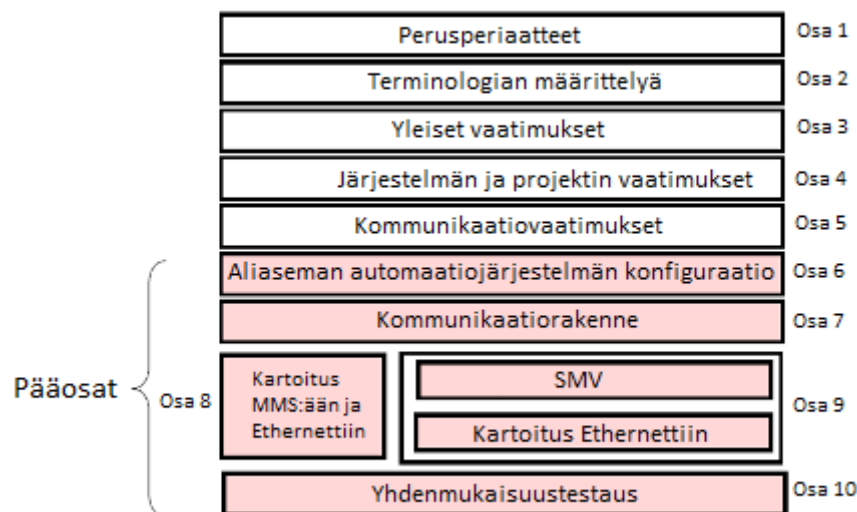
IEC61850-5:ssä on kommunikaatiovaatimukset funktioille ja laitemalleille. Siinä kerrotaan sähköasemien funktioiden taustaa sekä käsitellään loogisia nodeja. Vaikka tätä osaa ei olla määritelty yhdeksi pääosista, se on tärkeä osa, sillä siinä on lueteltu standardissa olemassaolevat loogiset nodet, ryhmiteltynä luokkiin (suojaus, mittaus, jne).

IEC61850-6 käsittelee IEC61850:n sähköautomaatiojärjestelmän konfiguraatiotakieltä. Toisin sanoen, tämä osa käsittelee SCL:ää.

IEC61850-7 dokumentoi standardin kommunikaatorakenteen. Seitsemäs osa jakautuu viiteen osaan. Nämä osat selvittävät sen, kuinka kommunikaatio toimii IEC 61850:ssa. Tärkeitä läpikäytäviä aiheita ovat mm. palvelut (ACSI) ja CDC:t.

IEC61850-8 ja IEC61850-9 kartoittavat standardin käyttämiä protokollia (SCSM). IEC61850-8 keskittyy MMS:ään (Manufacturing Message Specification) /14/, IEC61850-9 SV:en (Sampled Values) /15/.

Viimeisessä, kymmenennessä osassa selvitetään, kuinka standardia on testattu. Testaus on suoritettu melko perusteellisesti, kuten koko standardi on, ja siitä löytyy kymmeniä taulukoita erilaisille suoritetuille testeille. /16/ Kuvassa 3 IEC 61850:n kaikki osat.

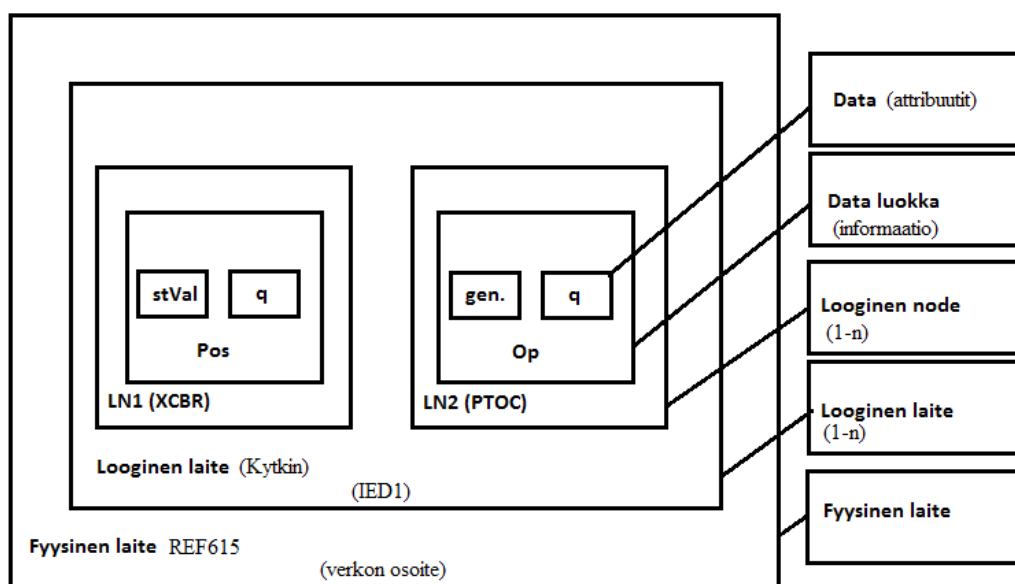


Kuva 3. IEC 61850:n rakenne. /17/

3.2 Datamalli

Sähköasemien toimintaan liittyy erilaisia toiminnallisuuksia, joita kutsutaan funktioiksi. Sähköasemalla on erilaisia funktioita, jotka ohjaavat sen toimintaa. Näitä funktioita ovat esimerkiksi suojausfunktiot. Suojausfunktioita taas on erilaisia. Esimerkki suojausfunktiosta olisi PTOC, ylivirran suojausfunktio. Ideana

IEC 61850:n datamallissa on, että jokin sähköaseman toiminnallisuuden osa, funktio, data tai laite, jaetaan osiin ja käsitellään tällä tavoin pilkottuna. Tämä jakaminen osiin on IEC 61850:n se osa, joka mahdollistaa laitteiden yhteensopivuuden. Se luo koko sähköasemalle ja sähköverkolle, jotka sitä käyttävät yhteisen ”kielen”, joka mahdollistaa kommunikaation asemien laitteiden kanssa. Tämä pilkkominen mahdollistaa myös sen, että samaa funktiota voidaan jakaa useammalle laitteelle. Tällaisia funktioita kutsutaan jaetuiksi funktioiksi. Kuva 4 havainnollistaa IEC 61850 -datan hierarkiaa.



Kuva 4. Datamallin eri tasot.

IED:n funktiot vastaanottavat erilaisia sisääntulosignaaleja sekä konfiguraatio- ja operaatioparametrejä ja tuottavat ulostulosignaaleja. Datamalli on standardisoitu tapa edustaa noita signaaleja ja parametrejä joita tarvitaan laitteiden väliseen kommunikaatioon. Näitä signaaleja ja parametrejä ilmaistaan loogisilla nodeilla. Loogiset nodet edustavat laitteen toiminnallisuuden pienimpiä osia. Ne edustavat funktioita, dataa tai laitteita, joita käytetään sähköjärjestelmissä.

Funktioit jakautuvat kolmeen tasoon. Prosessitason laitteet ovat yleensä etälaitteita ja sensoreita. Jotkin laitteet ovat tiettyä aseman kokonaisuutta varten omistettuja hallinta, suoja ja monitorointiyksiköitä. Asematason laitteisiin kuuluu aseman tietokone tietokantoinen, operaattorin työpiste, etäkommunikaation rajapinta,

ym.

Sen lisäksi, että laitteet ja funktiot on jaettu osiin loogisiksi nodeiksi loogiset nodet jakautuvat osiin dataobjekteiksi. Dataobjektit ovat funktioihin ja laitteisiin liittyvää tietoa. Esimerkiksi, johonkin suojafunktiioon voi liittyä SWI -looginen node, joka tarkoittaa katkaisijaa, ja tällä katkaisijalla on erilaisia ominaisuuksia joita ilmaistaan dataobjekteilla, kuten Pos eli asento.

Tämän lisäksi myös dataobjektit jakautuvat vielä pienempään tietoon, joita kutsutaan data-attribuuteiksi. Pos -dataobjektiin kuuluu esimerkiksi stVal -data-attribuutti (State Value), joka ilmaisee mikä on Pos -objektin tila. Eli kun yhdistetään nämä kolme, SWI.Pos.stVal, saadaan kytkimen tila, joka palauttaa tiedon joka on ON, OFF, INTERMEDIATE tai BAD-STATE, eli kytkin on joko päällä, pois päältä, siirtymässä tai virhetilassa.

3.2.1 Loogiset nodet

Loogiset nodet jakautuvat eri ryhmiin sen mukaan, mihin tarkoitukseen niitä käytetään sähköautomaatiossa. Jokaisella ryhmällä on kirjaintunnus, joka on noden nimessä ensimmäisenä. Näistä ryhmistä keskeisimmät on lueteltu taulukkoon 1.

Taulukko 1: Loogisten nodejen ryhmiä.

Ryhmä	Tunnus
Suojaus	P
Suojaukseen liittyvät funktiot	R
Hallinta	C
Rajapinta/loggaus/arkistointi	I
Automaatioprosessit	A
Mittausarvot	M
Tunnistus ja käyttäytyminen	L
Kytkimet	X
Monitorointi	S
Virranmuuntimet	T
Tehonmuuntimet	Y
Muut virtajärjestelmän laitteet	Z
Yleiset laitteet ja prosessit	G

Suojausryhmään P kuuluvat aseman suojausfunktiot. Esimerkiksi PTOC, eli AC time overcurrent protection eli ylivirran suojausfunktio, joka on funktio, jossa virran ylittäessä tietyn ennaltamääritetyn arvon tietyn ajan verran, rele ryhtyy suojaustoimiin.

Ryhmään R kuuluu suojauksessa tukena käytettyä dataa. Esimerkiksi RFLO eli Fault locator on funktio, joka paikantaa vian sijainnin.

Ohjausryhmään C kuuluu erilaisia hallintafunktioita. Esimerkiksi CSWI -nodea käytetään järjestelmän kytkinlaitteiston hallitsemiseen. Siis, vaikka kytkinlaitteistolle on määritetty oma ryhmänsä, X, niin sen hallitsemiseen käytetään erillistä hallitsemiseen tarkoitettua nodea.

Rajapinta-, loggaus- ja arkistointiryhmään kuuluu nimensä mukaisesti rajapintoihin liittyvää dataa. Esimerkiksi IHMI, human-machine interface, kuvaa paikallista hallitsemisrajapintaa joko bay -tasolla tai aseman tasolla. HMI:ien rooli vaihtelee sen mukaan, kuinka aseman rakentamisvaiheessa on päätetty. Tämän lisäksi ryhmään kuuluu kaksi muuta rajapintanodea ja IARC, arkistoinnissa käytetty node.

Automaatioprosesseihin kuuluu automaattisia funktioita, jotka pääosin pyrkivät pitämään jonkin arvon tietyissä rajoissa. AVCO, automatic voltage control, on node jota käytetään jännitearvon pitämisessä määritetyissä rajoissa. Toinen node joka kuuluu tähän ryhmään on AZVT, joka on node joka kääntää linjan pois päältä mikäli aliasema on ilman jännitettä jonkun määritetyn ajan verran.

Mittausarvojen ryhmään kuuluu nimensä mukaan erilaisia mittausarvoja. Ryhmän nodeista loogista nodea MMXU käytetään toiminnallisiin tarkoituksiin, kuten virran valvontaan, jolla mitataan arvoja kuten virta tai jännite. Nodea MMTR taas käytetään virran mittaamiseen kaupallisesta näkökulmasta.

Ryhmään L kuuluu vain yksi looginen node, LLN0 eli logical node device tai logical node 0. Tähän nodeen kuuluu yleistää data, joka liittyy laitteen fyysisiin ominaisuuksiin. Tällaista dataa on laatu (health), mode, käyttäytyminen (beh) ja tunnistus (name plate). Laitteella voi olla useita LLN0:ia joista kukin on omistettu

laitteen tietylle kokonaisuudelle. /8/

Kytkimien luokkaan X kuuluu sähköaseman kytkinlaitteet. Tähän luokkaan kuuluu kaksi loogista nodea: XCBB eli katkaisijat ja XSWI eli erottimet.

Monitorointiluokassa S on valvontaan käytettyjä loogisia nodeja. Siinä on kolme nodea: SIMS, SARC ja SPDC. Niitä kaikkia käytetään kaasun voimakkuuksien valvomiseen sähköasemassa.

Virranmuuntimiin kuuluu kaksi nodea, TCTR ja TVTR. Ensimmäinen on virranmuunnin ja toinen on jännitteenmuunnin.

Tehonmuuntimet, luokka Y, ovat nodeja joita käytetään tehonmuuntamisen prosessissa. Niitä ovat YPTR, YLTC, YEFN ja YPSH. YPTR on node varsinaiselle tehonmuuntimelle, muut nodet kuvaavat tehonmuunnoksessa käytettyjä apuvälineitä.

Luokka Z sisältää muita sähköasemaan liittyviä laitteita. Esimerkkejä: patteri (ZBAT), virtakaapeli (ZCAB), moottori (ZMOT) ja reaktori (ZREA).

Yleiset laitteet/prosessit pitävät sisällään funktion GGIO ja noden GSAL. GGIO viittaa aseman analogisiin I/O laitteisiin ja GSAL sisältää logeja aseman turvallisuuden toteutumisesta.

Edellämääritetyt loogisten nodejen luokat ovat ne olennaisimmat, joiden nodet on määritelty standardin 5. osassa, mutta luokkia on myös muitakin. Taulukossa 2 on lueteltuna muut ryhmät.

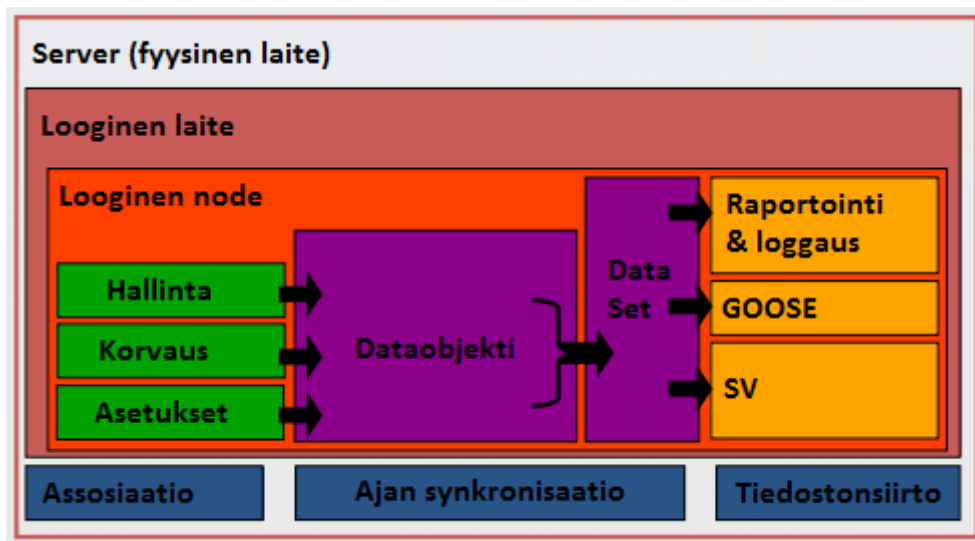
Taulukko 2: Loogisten nodejen ryhmiä.

Ryhmä	Tunnus
Varattu	B
Jaetut energialähteet	D
Varattu	E
Functional blocks	F
Vesivoima	H
Varattu	J
Mekaaniset laitteistot	K
Varattu	N
Varattu	O
Virran laadun tarkkailu	Q
Varattu	U
Varattu	V
Tuulivoima	W

Osa tunnuksista kuuluu ryhmään ”varattu”, mikä tarkoittaa, että tunnus on varattu tulevaisuutta varten, jossa sitä mahdollisesti käytetään joillekin nodeille. Tunnuksia W ja H käytetään vesi- ja tuulivoimajärjestelmien nodeissa, Q viittaa laatuun, K mekaaniseen ja ei-elektroniseen laitteistoon.

Sen lisäksi, että loogiset nodet voidaan jakaa näihin luokkiin, voidaan ne jakaa myös kolmeen eri tasoon. Ylimpään tasoon kuuluu aseman kokonaisuutta kuvaavat loogiset nodet. Tyypillisesti ryhmän I nodet ilmaisevat tätä tasoa, mutta myös A ja C voivat tehdä sitä. Keskimmäiseen tasoon kuuluu hallinta-, automaatio-, mittaus- ja suojafunktioita. Yleisimmät ryhmät tällä tasolla ovat P, C ja A, joskus myös X. Alimpaan, prosessi/kytkinlaitteistotasoon, kuuluu nodeja ryhmistä X, Y ja Z.

Loogiset nodet ovat datamallin keskiössä, mutta myös aseman muut osat huomioidaan datamallissa. Fyysinen laite eli server ilmaisee, mihin laitteeseen ollaan yhdistettynä, eli sen kommunikaatitiedot (IP ja portti). Looginen laite ilmaisee fyysiseen laitteeseen kuuluvaa yleistä tietoa (tunnistus, käyttäytyminen, ym.). Laitteen eri kokonaisuuksien esittämisen yhteydessä puhutaan dataseteistä, jotka ovat keino esittää relevanttia dataa yhden otsikon alla. Datasetit luetaan kuuluvaksi standardin palveluihin (ACSI). Kuvassa 5 IEC 61850:n datamalli.

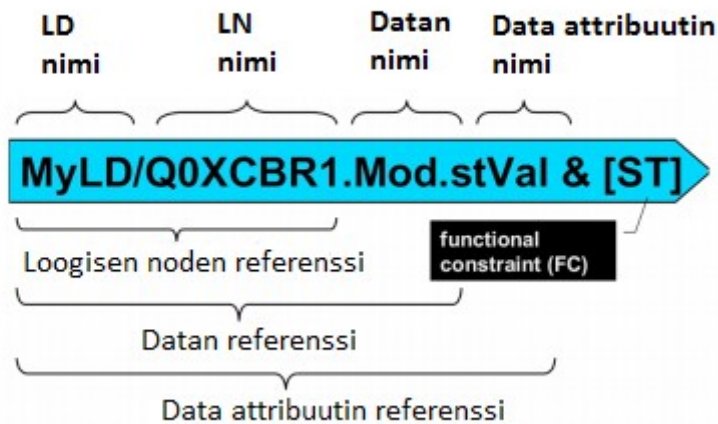


Kuva 5. Standardin datamalli. /2/

Kuvassa 5 assosiaatio viittaa serverin ja clientin väliseen kommunikaatioon, ajan synkronisaatio on tapa, jolla IED:t mittaavat aikaa.

3.2.2 Objektin nimeäminen

Objektin nimi IEC 61850:ssa määrittyy datamallin mukaisesti myös. Ensin määritellään loogisen laitteen nimi, seuraavaksi loogisen noden nimi, sen jälkeen dataobjektin nimi ja viimeisenä data-attribuutin nimi. Lopuksi voidaan määritellä myös functional constraint -arvo, joka kertoo minkälaisesta tiedosta on kyse, kuten onko kyseessä dataa, jota voidaan ainoastaan lukea. Kuvassa 6 objektin nimen rakenne IEC 61850:ssa.



Kuva 6. Objektin nimen rakenne. /18/

Loogisen laitteen nimen määrittelee joko valmistaja tai palveluntarjoaja, kuten myös loogisen noden ensimmäisen osan (kuvassa 6 Q0). Tätä ensimmäistä osaa käytetään sähköasemalla tunnistimena, missä kohtaa asemaa laite sijaitsee.

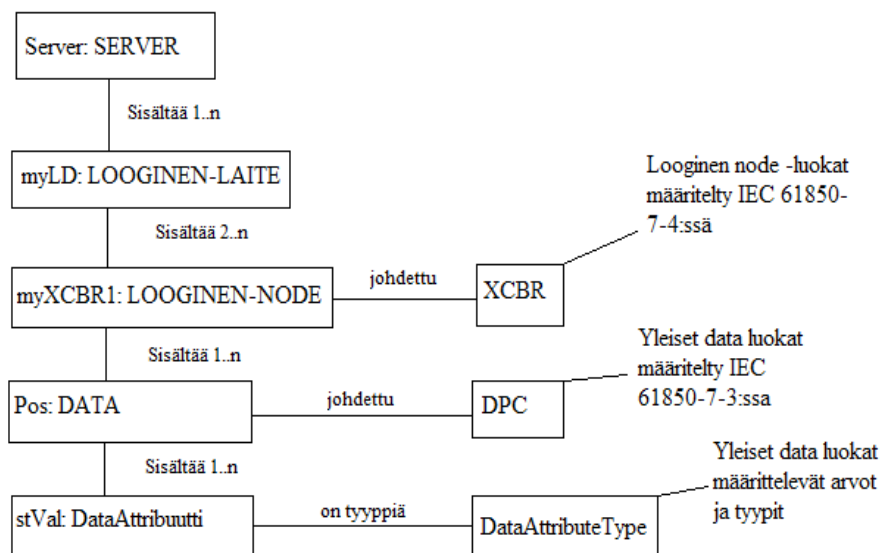
3.2.3 Yleiset dataluokat

IEC 61850:een kuuluu lisäksi luokkia, joita kutsutaan yleisiksi dataluokiksi (Common Data Class, CDC). CDC:t on määritelty standardin osassa 7-3, jossa määritellään kaikki yleiset dataluokat, sekä käydään läpi niiden data-attribuutit.

CDC:t ovat dataluokkia, jotka kuvaavat jotakin sähköaseman laitteen tai funktion osaa, ja niitä käytetään pohjana loogisten nodejen dataobjektien rakentamiseen. Ideana on, että tietyt dataobjektit sopivat yhteen tiettyjen yleisten dataluokkien kanssa ja aseman loogiset nodet käyttävät näitä luokkia omassa määrittelyssään. Esimerkiksi ACT, protection activation information, on CDC joka liittyy suojausten aktivoimiseen. Looginen node, joka esimerkiksi käyttäisi tätä luokkaa, olisi ylivirran suojausfunktio PTOC.

Toinen esimerkki CDC:stä on dataluokka, jonka kaikki kytkin -tyyppiset loogiset nodet (SWI, CBR) jakavat, eli kytkimien ja katkaisijoiden jakama Common Data Class, joka on nimetty DPC:ksi eli controllable double pointiksi, johon käytännössä viitataan yleensä dataobjektien tasolla lyhenteellä Pos eli position,

joka kuvaa kytkimen asentoa /13/. Kuvassa 7 on kuvattuna datamallin luokkakuva käyttäen esimerkkinä XCBR -loogista nodea ja sen dataa Pos, joka on johdettu DPC:stä /10/.



Kuva 7. Datamallin luokkakuva.

CDC:t pienentävät datan määrittelyn kokoa standardissa, koska useassa loogisessa nodessa ilmaantuvaa dataluokkaa ei ole ollut tarvetta määrittellä aina erikseen, vaan pelkkä viittaus on riittänyt, että node perii määritetyn CDC:n.

Esimerkki CDC:stä on dataluokka Measured Value (MV). MV:ä käytetään mittausarvon esittämiseen. Käytännössä tämä voi tarkoittaa esimerkiksi jotakin jännitelukemaa. Tuota lukemaa esitetään instMag ja mag -attribuuteilla, jotka ovat tyyppiä analoginen arvo. Signaalin laatua kuvaa q -attribuutti, ja t -attribuutti pitää sisällään aika-arvon. Lisäksi luokalla on M/O/C -arvo joka viittaa siihen, että vaaditaanko arvoa. Vaihtoehdot ovat mandatory (arvo vaaditaan), optional (arvo ei ole pakollinen) ja conditional (arvo vaaditaan tietyissä olosuhteissa). Mikäli ehtoja on, ne luetellaan luokan lopussa. Ehtoja voi olla useita. Tällöin ne erotellaan liukuluvulla (c1, c2, c3, jne.). /12/

Toinen esimerkki yleisestä dataluokasta on Sampled Value (SV). SV:ä käytetään näytteistetyn analogisen arvon esittämiseen. Kuten measured valuessa, instMag

ilmaisee analogista arvoa, q laatua ja t aika-arvoa /12/. Sampled value on sellainen arvo, jossa analoginen arvo on pilkottu tietyin väliajoin otetuksi näytteeksi, sampleksi, ja sitten lähetetty eteenpäin laitteelle, joka osaa muuntaa samplattyn tiedon takaisin analogiseksi ja tulkita sen. Samplattya tietoa hyödynnetään Sampled Values -SCSM:ssä, josta kerrotaan tarkemmin muualla tässä työssä.

Kolmas esimerkki on kuvassa 7 esiintynyt Controllable Double Point (DPC). DPC:ä käytetään kytkinlaitteiston asennon ilmaisemiseen. Käytännössä tähän dataluokkaan viitataan Pos -dataobjektin kautta. Sen data-attribuuteista ilmaistaan yleensä stVal, q ja t . StVal eli status value ilmaisee kytkinlaitteiston tilaa.

3.3 ACSI

ACSI eli Abstract Communication Service Interface on IEC 61850:n osa, joka tarkoittaa sen palveluja, eli mitä käsiteltävälle datalle tehdään. Palvelun katsotaan alkavan siitä, kun laite lähettää pyynnön toiselle laitteelle. Lähettävää laitetta kutsutaan clientiksi ja vastaanottavaa laitetta kutsutaan serveriksi. Vastaanottava laite lähettää vastauksen pyyntöön. Vastaus on joko positiivinen tai negatiivinen, eli toimitettiinko pyydetty toiminto onnistuneesti vai epäonnistui sen toteuttaminen.

Datamallin määrittäessä yhteisen kielen asemalle, ACSI määrittää sen, mitä datalla tehdään. ACSI tarjoaa toimintoja kuten datan lähettäminen ja vastaanotto, raportointi jonkin tapahtuman tapahtuessa, datan logaus, ym. Asemiin liittyy kahdenlaisia kommunikaatiopalveluita: Abstract Communication Services ja PICOM (Piece of information for communication). PICOM on kuvaus loogisesta yhteydestä kahden loogisen noden välillä. /5/

Standardin osassa 7-2 on määritelty eri palveluita, jotka tarjoavat pohjan informaation vaihtoon sähköasemassa. Näihin palveluihin kuuluu:

- Assosiaatio -palvelut – assosiaatio -palvelu tarkoittaa sitä, että kahden laitteen tai kokonaisuuden (client ja server) välille luodaan looginen yhteys. Nämä palvelut hoitavat laitteiden kommunikaatiot, yhteyksien purun ja kommunikaatiuongelmien hoidon.

- Get -palvelut – palvelut, joilla pyydetään dataa, jota halutaan nähdä. Näihin kuuluu esimerkiksi GetLogicalDeviceDirectory (hakee loogisen laitteen), GetLogicalNodeDirectory (hakee loogisen noden) ja GetDataValues (hakee arvot). Näitä palveluja käytetään tiedon valvomiseen.
- Set -palvelut – näitä palveluita käytetään hallitsemiskomennoissa (vrt. Control -luokka loogisissa nodeissa), parametrejen asettamisessa ja kuvausten kirjoittamisessa. Esimerkki: SetDataValues.
- Datasetit – datasetit ovat konsepti, jossa data ryhmitetään erilaisiksi ryhmiä tehokkaamman tiedonvaihdon saavuttamiseksi. Se on hyvin oleellinen palvelu IEC 61850:ssa. Yleensä dataset luodaan toisilleen relevanttejen loogisten nodejen yhdistelmällä. Datasetit ovat osittain valmiiksi määritettyjä: jokaisella loogisella nodella on oma datasettinsä. Nämä eivät aina vastaa järjestelmän tarpeita, joten datasettejä on mahdollista myös muuttaa sopimaan enemmän aseman tarpeisiin. Palveluntarjoajille vaatimuksena on, että he tarjoavat HMI:n (rajapinnan) datasettien muokkaamiseen.
- Report control – mahdollistaa raportoinnin hallitsemisen. Esimerkiksi, voidaan pyytää raportoimaan jonkin tapahtuman tapahtuessa, kuten jonkin datan arvon muuttuessa, tai signaalin laadun huonontuessa.
- Log control – logitiedostojen hallitsemispalvelu
- Substitution values – arvojen korvaamispalvelu. Jos halutaan uudelleenkirjoittaa jokin arvo aseman laitteessa, kutsutaan tätä palvelua.
- GSE messages – GSE eli Generic Substation Event -viestit ovat nopeita viestejä, joita käytetään aseman sisäisessä kommunikaatiossa viestittämään jostakin tapahtumasta.
- Select-Before-Operate control – kytkinlaitteiston turvamekanismipalvelu. Toimintaperiaatteena on, että alustaja (jokin IED, data-attribuuttien tasolla

viitataan attribuutissa origin, joka on tyyppiä Originator) lähettää kyselyn hallittavasta pisteestä (control point), vastaanottaja vastaa raportoimalla pisteen tilan, jonka jälkeen alustaja lähettää execute -komennon, jonka vastaanottaja suorittaa jos komento on lähetetty tietyn ennaltamääritetyn aika-arvon sisällä. Ideana on, että piste ”valitaan” ensin, ja sen jälkeen suoritetaan toiminto.

- Time management – ajan synkronisaatio nodejen välillä
- File transfer – tietojen siirto laitteiden välillä. Ideana on, että siirrettäviä tiedostoja ei kohdella dataobjekteina. Palvelun avulla on mahdollista ladata uusia ohjelmistoja laitteisiin.

Se, minkälaisia palveluita voidaan käyttää millekin data-attribuutille riippuu siitä minkälaista informaatiota kyseinen data-attribuutti edustaa. Tietoa, joka edustaa ainoastaan observoitavissa olevaa tietoa, voidaan ainoastaan lukea. Jännitelukemaa ei voida käskää olemaan jotain muuta kuin se on. Yleensä komennot kirjataan erilliseen control -attribuuttiin, jonka avulla IED päättää observoitavien tietojen mahdollisista muutoksista. Objektin muunneltavuus voidaan ilmoittaa sen nimen perässä olevassa Functional constraint -osassa.

ACSI:n tarjoamat palvelut voidaan kategorisoida yleisesti niiden tarjoamien ominaisuuksien pohjalta seuraavanlaisesti:

- datan käsittely
- laitteenhallinta
- tapahtumien raportointi ja logaaminen
- asetusrhmien hallinta
- laitteiden itsekuvailu
- tiedostojen siirto.

Datamalli kulkee mukana myös palveluissa. Tässä yhteydessä puhutaan luokkamalleista (class models), jotka ovat luokkia, jotka esittävät jotain IEC 61850 -datamallin tai palvelun osaa. Esimerkiksi looginen laite, dataset, assosiaatio tai GOOSE. Kuvassa 8 on kaikki ACSI:n malliluokat ja niihin liittyvät palvelut.

<p><u>GenServer</u> GetServerDirectory</p> <p><u>Association</u> Associate Abort Release</p> <p><u>GenLogicalDeviceClass</u> GetLogicalDeviceDirectory</p> <p><u>GenLogicalNodeClass</u> GetLogicalNodeDirectory GetAllDataValues</p> <p><u>GenDataObjectClass</u> GetDataValues SetDataValues GetDataDirectory GetDataDefinition</p> <p><u>DATA-SET</u> GetDataSetValues SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory</p> <p><u>SETTING-GROUP-CONTROL-BLOCK</u> SelectActiveSG SelectEditSG SetEditSGValue ConfirmEditSGValues GetEditSGValue GetSGCBValues</p> <p><u>REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK</u> BUFFERED-REPORT-CONTROL-BLOCK: Report GetBRCBValues SetBRCBValues UNBUFFERED-REPORT-CONTROL-BLOCK: Report GetURCBValues SetURCBValues</p>	<p><u>LOG-CONTROL-BLOCK</u> GetLCBValues SetLCBValues QueryLogByTime QueryLogAfter GetLogStatusValues</p> <p><u>Generic substation event – GSE</u> GOOSE SendGOOSEMessage GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues</p> <p><u>Transmission of sampled values</u> MULTICAST-SAMPLE-VALUE-CONTROL-BLOCK: SendMSVMessage GetMSVCBValues SetMSVCBValues UNICAST-SAMPLE-VALUE-CONTROL-BLOCK: SendUSVMessage GetUSVCBValues SetUSVCBValues</p> <p><u>Control model</u> Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate</p> <p><u>Time & time synchronization</u> TimeSynchronization</p> <p><u>FILE transfer</u> GetFile SetFile DeleteFile GetFileAttributeValues</p>
---	--

Kuva 8. ACSI:n malliluokat ja niiden palvelut.

Jos esimerkiksi haluttaisiin päästä käsiksi jonkin IED:n dataobjekteihin, yhteyspisteen määrittämisen jälkeen käytettäisiin Association -mallia assosiaation aloittamiseen. Tämän jälkeen voitaisiin hakea GenLogicalDeviceClassilla looginen laite, tai sitten GenLogicalNodeClassilla hakea kaikki loogiset nodet (GetAllDataValues). Sen jälkeen käytettäisiin GenDataObjectClass -mallin

GetDataValues -palvelua. Lopuksi voitaisiin käyttää DATA-SET -mallin CreateDataSet -palvelua datasetin luomiseen.

Standardin palvelut, CDC:t ja loogiset nodet muodostavat hierarkisen kokonaisuuden. Ylimmällä tasolla on loogiset nodet, jotka käyttävät yleisiä dataluokkia joidenkin objektiensa muodostukseen, kun taas CDC:t käyttävät ACSI:ssa määriteltyä GenCommonDataClass -mallia omassa määrittelyssään.

3.3.1 Functional constraint

IEC 61850:ssa objekteilla on arvo, jota kutsutaan functional constraint -arvoksi. Tyypillisesti arvo ilmaistaan objektin perässä hakasuluissa, esim. Pos.stVal [ST]. Functional constraintit ovat tapa kategorisoida data-attribuutteja eri ryhmiin. Tällä hetkellä olemassa olevat functional constraintit on lueteltuna taulukossa 3.

Taulukko 3: Functional constraintit.

Lyhenne	Merkitys (engl.)	Sallitut palvelut
ST	Status information	read, substitute, report, log
MX	Measurands	read, substitute, report, log
SP	Setting	read, write, report
SV	Substitution	read, substitute
CF	Configuration	read, write, report
DC	Description	read, write
SG	Setting group	
SE	Setting group editable	
SR	Service response	
OR	Operate received	
BL	Blocking	
EX	Extended definition	
XX	Representing data attributes	

ST on functional constraint, jota käytetään tilainformaatiota ilmaisevissa data-attribuuteissa. Esimerkiksi XCBR.Pos.stVal omaa functional constraintin ST, koska se ilmaisee katkaisijan kytkimen tilaa. Sallittuja palveluja data-attribuuteille, joilla on tämä functional constraint on read, substitute, report ja log, eli siihen ei voida käyttää write -palveluita.

MX on functional constraint, jota käytetään mitta-arvoissa. MX -functional constraintin omaavat data-attribuutit edustavat mitattavaa (analogista) arvoa

sähköasemassa. Sallittuja palveluja ovat read, substitute, report ja log.

SP -functional constraintin data-attribuutit edustavat asetusparametrejä ja niitä voidaan lukea, kirjoittaa ja raportoida. SV viittaa korvausarvoihin, eli dataan jota käytetään korvaamaan jokin vanha arvo. Datan palvelut ovat read ja write. CF on functional constraint, jota käytetään konfiguraatioinformaatiossa. Sitä voidaan lukea ja kirjoittaa.

DC tarkoittaa kuvausinformaatiota. Nykyajan IED:ien ja IEC 61850:n ominaisuuksiin kuuluu laitteiden itsekuvaileminen. DC -functional constraintin data liittyy erityisesti tähän ominaisuuteen. Palvelut read ja write.

SG:lla viitataan Setting group -dataan. Loogiset laitteet, jotka implementoivat SGCB eli Setting group control block -luokka pitävät sisällään useita ryhmitettyjä arvoja kaikista data-attribuuteista functional constraintilla SG. Tämän functional constraintin omaaviin data-attribuutteihin ei voi käyttää write -palveluita. SE on muutoin sama kuin SG, mutta arvoja voi muokata SGCB:n palveluilla.

Datan raportoinnissa puhutaan trigger option eli TrgOp -attribuutista. Se on attribuutti, jolla määritetään ne ehdot, jotka aiheuttavat raportin lähettämisen tai logimerkinnän tekemisen. Trigger optioneita on kolme: dchg, qchg ja dupd. Dchg on lyhenne sanoista data change ja tarkoittaa sitä, että raportointi tai loggaus tehdään kun kyseessä olevan data-attribuutin arvossa tapahtuu muutos. Qchg eli quality change aiheuttaa raportoinnin kun data-attribuutissa q, eli laatu, tapahtuu muutos. Dupd eli data value update tarkoittaa sitä, että raportointi suoritetaan kun data-attribuutin arvo päivitetään.

Functional constrainteihin liittyy olennaisesti myös data, jota kutsutaan Functional constrained dataksi, lyhennettynä FCD:ksi. Jonkin dataobjektin data-attribuutteja, jotka on järjestelty kokoelmaksi, ja joilla on sama functional constraint kutsutaan functional constrained dataksi. Viittausta yksittäiseen data-attribuuttiin, joka kuuluu tietyn functional constraintin omaavaan dataobjektiin kutsutaan functional constrained data-attribuutiksi (FCDA).

Datasetit ovat ryhmä objektiviittauksia functional constraint dataan ja data-

attribuutteihin, jotka on järjestelty yhdeksi kokonaisuudeksi helppokäyttöisyyden vuoksi. Näitä objektiviihtauksia kutsutaan datasetin jäseniksi. Datasettien koostumus on tiedossa sekä clientillä että serverillä, joten kommunikaatiossa ainoastaan datasetin nimi ja dataobjektien ja attribuuttien nykyinen arvo täytyy välittää. Dataobjekteihin ja attribuutteihin voi olla viittaus useassa datasetissä serverissä. Datasettejä konfiguroidaan ja luodaan CreateDataSet -palvelun avulla.

3.3.2 CST

Palveluita voidaan seurata Common service tracking (CST) avulla. Se on yleinen dataluokka palveluiden seuraamista varten, jolla voidaan seurata kaikkia palveluita, mukaan lukien kielteisen vastauksen saaneita. Sen attribuutit on kuvattuna kuvaan 9. Seuraamista varten palvelut on jaettu kolmeen ryhmään: Control block eli ohjauslohko -palvelut, komentoihin liittyvät palvelut ja muut palvelut. Luokka periytyy muille seurantalukille. /11/

CST luokka						
Attribuutin nimi	Attribuutin tyyppi	FC	TrgOp	r/w	Arvo/arvon vaihteluväli	M/O/C
DataName	Inherited from Data Class					
Kuuluu vain CST:hen						
objRef	ObjectReference	SR	dupd	r	Viittaus objektiin, jota käytetään seuraamisessa: joko control block, jota käytetään tai control objekti, jota hallitaan.	M
serviceType	ServiceType	SR		r	Seurattavan palvelun tyyppi.	M
errorCode	ServiceError	SR		r	serviceType:ssä määritettyyn palveluun liittyvä virhe. Onnistuneilla palveluilla arvo no-error.	M
originatorID	OCTET STRING64	SR		r	Palvelun aloittaja.	O
t	TimeStamp	SR		r	TimeStamp ajanhetkelle, jolloin palvelu on suoritettu	M
d	VISIBLE STRING255	DC		r		O
dU	UNICODE STRING255	DC		r		O
cdcNs	VISIBLE STRING255	EX		r		AC_DLNDA_M
cdcName	VISIBLE STRING255	EX		r		AC_DLNDA_M
dataNs	VISIBLE STRING255	EX		r		AC_DLN_M

Kuva 9. Palveluiden seuraamiseen käytetty CST -luokka.

Attribuutti objRef viittaa siihen objektiin, jota palvelu käyttää. Ohjauslohkojen tapauksessa se on viittaus siihen ohjauslohkoon, jonka attribuutteja seurataan. Control -palveluiden kanssa se on viittaus hallittavaan dataobjektiin. Attribuutti

originatorID tunnistaa clientin, joka aloitti palvelun.

Ohjauslohkot on mallinnettu samankaltaisesti, kuin dataobjekti -luokat. Keskeisenä erona niiden välillä on, että dataobjekteja käytetään luomaan rajapinta ohjelmatason funktioille, kun taas ohjauslohkoja käytetään kommunikaatiopalveluiden konfiguroimiseen.

IEC 61850:ssa on eri tyyppisiä ohjauslohkoja erilaiselle informaatiolle. Ohjauslohkoja on puskuroidulle ja puskuroimattomalle raportoinille, logitiedostojen ohjauslohkoille, GOOSElle, Multicast SV:lle, Unicast SV:lle ja asetusrhymille. Niitä voidaan seurata myös seurantapalveluiden avulla. Yleiset dataluokat, joita tähän käytetään ovat:

- buffered report Tracking Service eli BTS
- unbuffered report Tracking Service eli UTS
- log control block Tracking Service eli LTS
- GOOSE control block Tracking Service eli GTS
- MSVCB Tracking Service eli MTS
- USVCB Tracking Service eli NTS
- SGCB Tracking Service eli STS.

Nämä kaikki luokat perivät edellämääritellyn yleisen dataluokan CST:n.

IEC 61850:ssa raportoinnilla tarkoitetaan sitä, että serverit voivat lähettää dataa eri tapahtumien yhteydessä ilman erillistä kyselyä clientiltä. Se, mitä dataa lähetetään ja mitkä tapahtumat aiheuttavat raportoinnin konfiguroidaan osissa, joita kutsutaan raportoinnin hallintakappaleiksi (report control block) eli RCB:ksi.

Standardissa raportointi on eroteltu kahteen erilaiseen tyyppiin: puskuroituun ja puskuroimattomaan raportointiin. Puskuroidussa raportoinnissa palvelin puskuroi raportit siltä varalta, että yhteys clientiin katkeaa. Tällä tapaa raportit voidaan

lähettää, kun yhteys clienttiin on saavutettu uudestaan. Puskuroitu raportointi konfiguroidaan sille varatuissa puskuroidun raportoinnin ohjauslohkoissa (BRCB) ja puskuroimaton raportointi konfiguroidaan puskuroimattoman raportoinnin ohjauslohkoissa (URCB). /11/

RCB:t sijaitsevat serverin datamallin loogisissa nodeissa, ts. palvelimen ICD/SCL-tiedostossa. Täten niihin viitataan muodossa LooginenLaite/LooginenNode.RCBNimi. RCB voi viitata vain datasetteihin, jotka sijaitsevat samassa loogisessa nodessa kuin se itse, tosin muissa loogisissa nodeissa sijaitsevaa dataa voidaan monitoroida, koska datasetin jäsenet voivat sijaita useissa loogisissa nodeissa. Monitoroitavaa datasettiä voidaan vaihtaa ”lennosta”, clientin ollessa päällä. Yksi RCB voi raportoida vain yhdelle clientille kerrallaan.

Seuraavat tapahtumat voivat saada aikaan raportoinnin:

- client lähettää kyselyn.
- jokin RCB:n monitoroiman datasetin data-attribuutti vaihtuu tai päivitetään.
- ajastetun raportoinnin ollessa päällä raportti lähetetään määritellyn ajan välein.

Raportointiohjauslohkoon kuuluu seuraavat tietokentät:

- Konfiguraatiokentät. Voidaan kirjoittaa vain, jos RCB ei ole toisen clientin varaama tai käytössä:
 - RptId: RCB:n tunnistin
 - DataSet: viittaus RCB:n käyttämään datasettiin
 - OptFlds: bittijono, jossa jokainen bitti määrittää, onko jokin valinnainen kenttä (optional field) käytössä viitatussa RCB:ssä. Olemassaolevat valinnaiset kentät ovat sequence-number, report-

timestamp, reason-for-inclusion, data-set-name, data-reference, buffer-overflow, entryID, segmentation ja conf-revision.

- BufTm: määrittää aikavälin millisekunnissa, joka odotetaan seuraavaa tapahtumaa. Kaikki tapahtumat, jotka raportoidaan tämän aikavälin aikana lähetetään yhdessä raportissa.
- TrgOps: määrittää mitkä tapahtumat aiheuttavat raportoinnin. Mahdolliset tapahtumat ovat data arvon muutos (dchg), laadun muutos (qchg), arvon päivitys (dupd), kokonaisuus (kokonaisuusraportit pitävät sisällään datasetin kaikkien jäsenten arvot), yleinen kysely (GI).
- IntgPd: määrittää aikavälin, jonka välein lähetetään kokonaisuusraportti
- PurgeBuf: kuuluu vain BRCB:hin
- EntryID: kuuluu vain BRCB:hin
- Informaatiokentät, voidaan ainoastaan lukea:
 - SqNum: RCB:n senhetkinen jaksonumero. Raportointi aiheuttaa numeron kasvattamisen yhdellä.
 - ConfRev: määrittää kuinka usein data-arvoa on muutettu
 - Owner: sen clientin IP, joka on varannut RCB:n
 - TimeOfEntry: kuuluu vain BRCB:hin
- Toiminnalliset kentät, voidaan lukea ja kirjoittaa suorittamaan funktioita:
 - Resv: määrittää onko RCB varattu
 - ResvTms: kuuluu vain BRCB:hin

- RptEna: määrittää, onko raportointi otettu käyttöön
- GI: yleinen kysely. Arvon asettaminen todeksi aiheuttaa raportin lähettämisen clientille. Raporttiin sisältyy datasetin kaikki arvot.

Tietokentät RptID, OptFlds, SqNum, TimeOfEntry, DataSet, BufOvfl (puskurin ylittämisestä seuranneet tiedonmenetykset), EntryID, ConfRev, SubSqNum (jos raportti on pitkä, se jaetaan osiin, osilla on sama SqNum mutta yksilöllinen SubSqNum), MoreSegmentsFollow (viestii, että jaetussa raportissa on vielä tulossa lisää osia), Inclusion Bitstring (bittijono, joka määrittää mitkä datasetin arvot kuuluvat raporttiin), Data-references (datasetin jäsenten referenssit), Values (raportoidun datasetin jäsenten arvot) ja Reason Codes (kertoo syyn, miksi arvo raportoitui) kuuluvat aina raporttiin.

Client, joka haluaa vastaanottaa raporteja ensin yrittää varata RCB:n. Tämän jälkeen, mikäli varaaminen onnistuu, RCB konfiguroidaan tarpeita vastaavaksi. Lopuksi raportointi otetaan käyttöön.

3.4 SCSM

Kommunikaatioteknologiat kehittyvät ajan mukaan, ja IEC 61850 kehittyy niiden mukana protokollien kartoituksen, eli Specific Communication Service Mappingin (SCSM), avulla. Protokolla voidaan mieltää laitteiden väliseksi yhteiseksi kieleksi, jonka avulla kaksi tai useampi laitetta kommunikoi keskenään. Se määrittää sen, kuinka IED:n data, pyynnöt ja vastaukset ilmaistaan. Tällä hetkellä kaikki IEC 61850:n käyttämät protokollat perustuvat Ethernetiin.

IEC 61850 käyttää kolmea protokollaa: MMS:ää, SV:ä ja GOOSEa. Nämä protokollat voivat toimia TCP/IP:n tai LAN:n yli käyttäen korkeanopeuksista Ethernetiä saavuttaakseen relesuojaukseen vaaditun alle neljän millisekunnin reaktioajan. Neljäs protokolla Web palveluille on kehitteillä. Ennen sähköasemissa on käytetty kuparijohdottamista (hard-wiring) laitteiden välillä, mutta näiden uusien protokollien myötä ollaan siirrytty käyttämään Ethernetiä, ja johdottamisestakin on tullut sitä mukaa kevyempää. Kehitys sähköasemissa on ollut ja on edelleenkin koko ajan siirtyminen raskaasta johdottamisesta

kevyempään kommunikaatioprotokollien kehittyessä.

3.4.1 GOOSE

GOOSE eli Generic Object Oriented Substation Event on nopea, reaaliaikainen, tapahtumien ohjaama mekanismi, jossa kaikenlaisen formaatin data lyödään yhteen datasettiin ja välitetään neljän millisekunnin ajassa. Generic viittaa siihen, että voidaan käyttää mitä vain dataa, Object Oriented viittaa siihen, että data on IEC 61850 datamallin mukaista, Substation viittaa siihen, että koko asema näkee datan ja Event viittaa siihen, että lähettäminen perustuu aseman tapahtumiin (ts. se on automatisoitu). GOOSEa käytetään datan välittämiseen aseman sisällä samantasoisten laitteiden välillä. Se kykenee kaikenlaisen tiedon välittämiseen IED:ien välillä.

GOOSE määrittellään protokollan osissa 7-2 ja 8-1. Idea GOOSEssa (kuten myös SV:ssä) on, että se toimii Ethernetin päällä, yleensä switched Ethernet -verkossa, nopean tapahtumien vaihdon laitteiden välille saavuttaakseen. Koska se on suoraan Ethernetin päällä, se toimii vain paikallisissa verkoissa eikä ole IP-reitittimien reititettävissä. GOOSE on rakenteeltaan hyvin yksinkertainen ja sitä voivat käyttää paljon yksinkertaisemmat laitteet kuin esimerkiksi MMS-palveluille vaaditut laitteet voivat käyttää. GOOSE voidaan konfiguroida staattisesti ja dokumentoida ICD -tiedostoon tai konfiguroida dynaamisesti manipuloimalla Goose Control Blockia MMS:n kautta. Tarkoitus GOOSEssa on, että data lähetetään dataseteissä ja tapahtumat voidaan jakaa asemanlaajuisesti käyttäen multicastia.

Tiedonsiirron tasolla GOOSE käyttää (tai voi käyttää) seuraavanlaisia mekanismeja datasiirron nopeuden ja luotettavuuden varmistamiseksi:

- GOOSE -data sulautetaan Ethernet data paketeiksi, jonka jälkeen data joko multicastataan tai broadcastataan publish-subscribe -periaatteella.
- GOOSE käyttää VLAN:ia omatakseen erillisen virtuaalisen verkon samassa fyysisessä verkossa ja asettaa sopivan prioriteettitason viesteille

- Datan uudelleensiirtomekanismit – sama GOOSE -viesti lähetetään uudelleen vaihtelevin välein. Muutokset GOOSE dataset -elementissä aiheuttavat olemassaolevan GOOSE -viestin välityksen lopettamisen. Tilanumero GOOSE -protokolla tunnistaa onko GOOSE -viesti uusi vai uudelleenvälitetty viesti
- GOOSE -viestit on suunniteltu olemaan brändiriippumattomia. IEC 61850:aa täysin tukevat IED:t eivät vaadi juuri tietynlaisia kaapeleita tai algoritmeja

3.4.2 MMS

MMS eli Manufacturing Message Specification on kansainvälinen, ISO 9506:ssa määritelty standardi, jonka tarkoituksena on välittää reaaliajassa prosessidataa verkossa olevien laitteiden kanssa. IEC 61850:ssa se on valjastettu vertikaaliseen kommunikaatioon, eli hierarkisesti eri tasolla olevien laitteiden väliseen kommunikaatioon. MMS on GOOSEa paljon raskaampi ja monimutkaisempi, GOOSEn ollessa paljon riisutumpi protokolla, joka lähettää vähemmän tietoa kerrallaan.

3.4.3 SV

SV eli Sampled Values on toinen horisontaalisessa kommunikaatiossa käytettävä protokolla. Tällä hetkellä sitä ei juurikaan käytetä, mutta se tekee tuloaan sähköasemien kommunikaatiossa ja on tulevaisuuden protokolla, joka tulee hiljalleen saavuttamaan vankan aseman sähköasemissa. Kuten GOOSE, myös SV toimii suoraan Ethernetin päällä. SV, kuten GOOSE, toimii publisher/subscriber -periaatteella. Subscribereita (ts. client) voi olla yksi (unicast SV) tai useampia (multicast). Multicastia käytettäessä käytetään multicast -osoitteita ja subscriberit konfiguroidaan kuuntelemaan määritettyä multicast -osoitetta. Subscriberin osoite määritetään Sampled Values -ohjauslohkossa (SVCB).

Idea SV:ssä on, että analogisia arvoja samplataan tietty määrä pulssin aikana, ja lähetetään samplattu arvo eteenpäin IED:lle, joka osaa tulkita samplatyn arvon. Hyöty mikä tässä on se, että sähköasemalla ei tarvitse vetää kuparijohdotusta

sähkölinjoilta releille asti, vaan esimerkiksi sähkölinjan lähellä on laite, jonka luo tarvitsee laittaa lyhyemmin johtoa, ja tältä laitteelta sitten lähtee Ethernet -johto releille. Tämä sähkölinjojen luona oleva laite toimii SV -julkaisijana aseman muille releille.

3.5 SCL

Järjestelmän suunnittelu alkaa yleensä ennen kuin mitään fyysistä kokonaisuutta on olemassa. Tämän lisäksi nykyaikaiset IED:t pystyvät suoriutumaan useista eri tehtävistä. Tämä ei kuitenkaan tarkoita välttämättä, että kaikkia mahdollisia toimintoja pystytään suorittamaan samaan aikaan, minkä johdosta on tarpeellista määrittää mitä laite voi tehdä. Konfiguraatitietojen tallentamiseen on IEC 61850:ssa määritelty oma XML:ään perustuva konfiguraatiokieli, substation configuration language eli SCL. Tällä hetkellä SCL:stä on olemassa kaksi versiota, jotka eivät ole yhteensopivia, joten asemalla voi olla tarpeellista käyttää samaa SCL:n versiota kaikkialla. Taulukossa 4 on SCL:n eri tiedostotyypit merkityksineen.

Taulukko 4: SCL:n tiedostotyypit.

Tiedostotyyppi	Merkitys (engl.)
.ssd	System specification description
.icd	IED capability description
.iid	Instantiated IED description
.scd	System description
.cid	Configured IED description
.sed	System exchange description

Taulukossa 4 on lueteltuna SCL:n eri tiedostotyypit, .ssd, .icd, .iid, .scd, .cid ja .sed, sekä niiden merkitykset. Se, mihin tarkoitukseen SCL -tiedostoa käytetään määrää sen, mitä tiedostotyyppiä käytetään.

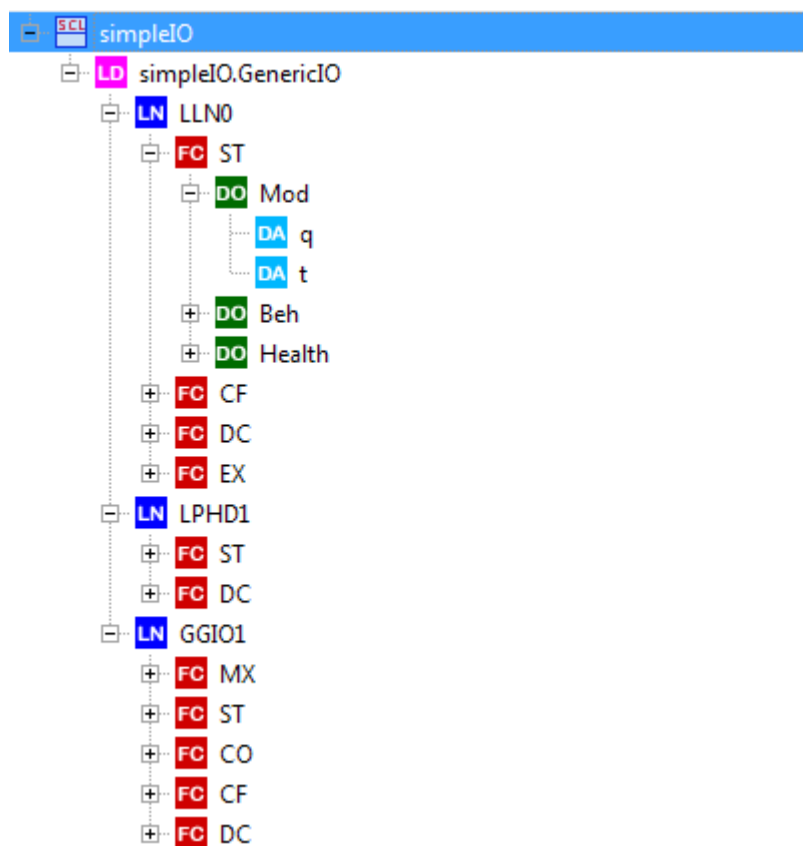
1. ICD (IED capability description) -tiedostoa käytetään kuvaamaan IED:n täyttä toiminnallisuutta. Laitteen valmistaja luo tämän tiedoston laitteelle, se kuvaa siis laitteen nk. tehdasasetukset. Tiedostoon kuuluu IED -osio, mahdollinen kommunikaatio-osio ja mahdollinen aliasemaosio.

2. SSD (System specification description) -tiedosto sisältää koko ala-aseman määrittelyn (single line diagram ja loogiset nodet). Tiedostossa on osio ala-asemalle, datatyyprien mallinteille ja loogisten nodejen tyyppien määrittelyille, mutta IED -osio ei ole tarpeellinen.
3. SCD (Substation configuration description) -tiedostossa kuvaillaan ala-aseman yksityiskohdat. Siihen sisältyy ala-asema, kommunikaatio, IED ja datatyypimallinne osio. SSD ja ICD -tiedostot tekevät yhteistyötä SCD -tiedoston luomiseen.
4. CID (Configured IED description) on tiedostotyyppi jota käytetään apuna IED:n konfigurointityökalun ja IED:n kommunikoimisessa. Se on periaatteessa SCD -tiedosto, johon on jätetty vain sellainen tieto, jota käytettävä IED tarvitsee. Siinä on pakollinen kommunikaatio-osio käytettävän IED:n määrittelyä varten.
5. IID (Instantiated IED description) -tiedosto määrittää IED:n konfiguraation projektia varten ja sitä käytetään datavaihdon mallina IED:n konfiguroinnin ja järjestelmän konfiguroinnin välillä. Tähän tiedostoon kuuluu ainoastaan konfiguroitavan IED:n data: yksi IED -osio, kommunikaatio-osio kommunikaatioparametreineen, IED:n datatyypimallinteet ja mahdollisesti aliasemaosio, jossa loogiset nodet on sidottuna SLD:n.
6. SED (System exchange description) -tiedosto on tiedosto, jota eri projektien järjestelmänkonfiguroijat vaihtavat keskenään. Se kuvaa projektin rajapinnat, joita toinen projekti voi käyttää. /9/

IEC 61850 -kommunikaation konfiguraatio koostuu useista vaiheista. Jotkut vaiheista pitää tehdä ennen toisia vaiheita, mutta yleisesti ottaen ne voidaan tehdä missä tahansa järjestyksessä. Riippuen siitä, miten konfiguraatioprosessia lähestytään, puhutaan ylhäältä alas -lähestymistavasta, alhaalta ylös -lähestymistavasta tai sekalaisesta lähestymistavasta. Ensiksi mainitussa konfiguraatioprosessi alkaa sähköaseman yleisistä piirteistä ja liikkuu kohti

yksityiskohtia. Toiseksi mainitussa toimitaan päinvastoin, aloittamalla yksityiskohtien määrittelyllä ja antaen niiden hiljalleen muodostaa suurempia kokonaisuuksia. Viimeiseksi mainitussa lähestymistavassa käyttäjän ei tarvitse seurata kumpaakaan edellisistä, vaan hän sekoittaa lähestymistapoja niin kuin katsoo parhaaksi.

Kuvassa 10 on kuvakaappaus IEDModeler -ohjelmaan avatusta SCL -tiedostosta.

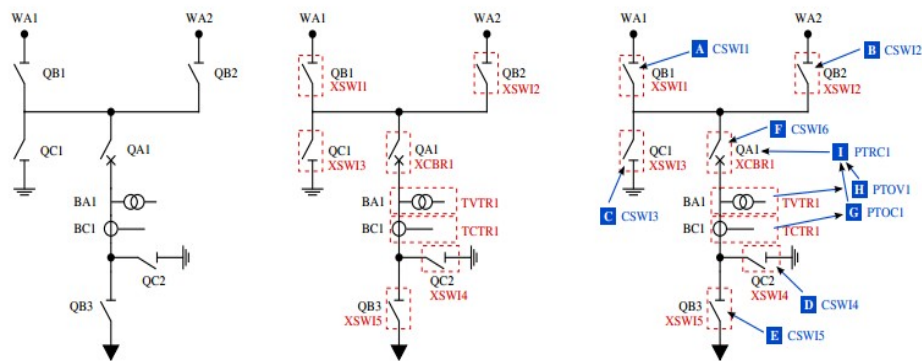


Kuva 10. SCL -tiedoston data.

3.5.1 SCL -työkalut

SCL:n liittyy neljä työkalua. Ensimmäisen on järjestelmän määrittelytyökalu. Sitä käytetään SLD:n eli single-line diagramin piirtämiseen. Single-line diagram on yksinkertaistettu kuvio sähköjärjestelmästä. Kuvassa 11 on esimerkki SLD:stä. Ylhäältä alas -lähestymistavassa tämä on ensimmäinen askel konfiguraatioprosessia. Toinen työkalu on järjestelmän konfigurointityökalu. Sitä

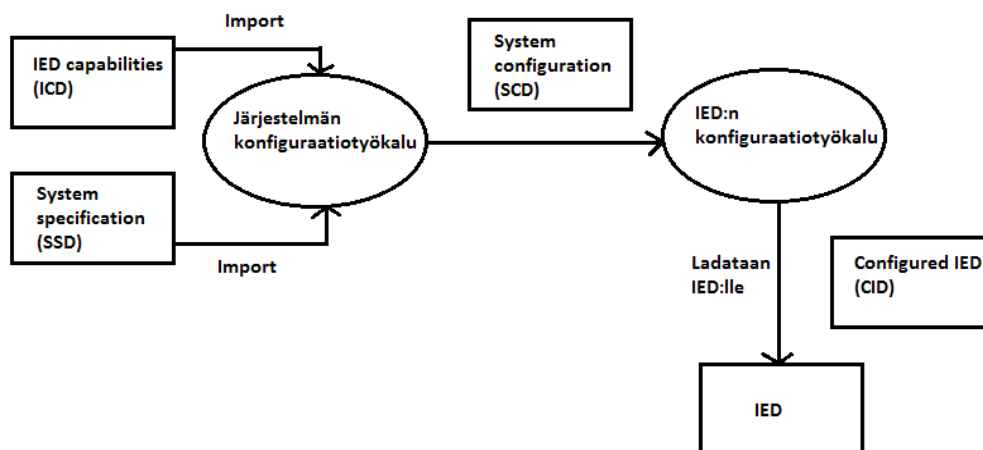
käytetään laitteiden välisen IEC 61850 -kommunikaation asentamiseen. Kolmas työkalu on IED:n konfigurointityökalu, jota käytetään IED:n yksityiskohtaiseen konfiguraatioon. Kukin IED:n konfigurointityökalu sopii vain tietynlaiseen IED:een, täten voi olla tarpeellista käyttää useita IED konfiguroijia aseman pystyttämässä. IED configurator toolit hoitavat kaiken konfiguraatitiedon, jota ei ole standardisoitu. Viimeinen työkalu on dokumentaatiotyökalu, minkä tehtävä on automatisoida laitteiston, ohjelmiston ja parametrien dokumentaatio.



Kuva 11. Single line diagram (SLD). /18/

Ensimmäisessä kuvassa vasemmalta SLD:n yksinkertaisin muoto. Toisessa mukana prosessitason loogiset nodet. Kolmannessa mukana aseman loogisia nodeja.

SCL -tiedoston konfigurointiprosessi etenee siten, että ICD ja SSD -tiedostot importoidaan ensin järjestelmän konfigurointityökalulle. Siitä syntyvä SCD -tiedosto importoidaan IED:n konfigurointityökalulle, jossa suoritetaan myös suunnitelun viimeistely. Syntyvä CID -tiedosto voidaan ladata laitteille valmistajan määrittämällä konfiguraatiotyökalulla. Kuvassa 12 on kuvattuna SCL -tiedoston konfiguraatioprosessi.



Kuva 12. SCL -tiedoston konfiguraatioprosessi.

SCL -tiedostoon kuuluvat seuraavat osat:

1. Header eli otsikko määrittää version ja muita perustietoja
2. Substation määrittää aseman eri kokonaisuudet, kuten laitteet, kytkennät ja muita toiminnallisuuksia. Myös toiminnallisuutta kuvaavat loogiset nodet mainitaan.
3. Communication määrittää kommunikaatiopisteet eri IED:eille koko systeemissä
4. IED -osio määrittää koko IED:n laitteen konfiguraation. Se kuvaa mitä dataa IED välittää raporteina ja GOOSE -viesteinä ja mitä GOOSE/GSSE -dataa se on konfiguroitu vastaanottamaan.
5. DataTypeTemplates -osio määrittää eri loogiset laitteet, loogiset nodet, datan ja muut yksityiskohdat eroteltuina eri osioihin. Se noudattaa IEC 61850 -standardin datamallinnusta. Se jaetaan edelleen aliosioihin LNodeType, DOType, DAType ja EnumType.

SCL voidaan jakaa myös toisella tapaa neljään eri osioon seuraavanlaisesti:

1. Substation subsection: määrittää SLD:n ja miten se sitoutuu loogisiin nodeihin ja myös loogisten nodejen sijoitus IED:en
2. Communication section: kuvaa kommunikaatioyhteydet IED:en välillä
3. IED section: kuvaa IED:en konfiguraation ja mitkä loogiset nodet se jakaa toisten IED:en kanssa
4. LNTYPE section: määrittää mitkä dataobjektit otetaan mukaan loogisiin nodeihin.

4 KIRJASTOT

4.1 Yleistä

IEC 61850:lle on olemassa muutamia kirjastoja ohjelmointia varten. Isoilla yhtiöillä on pääasiassa omat kirjastonsa, mutta muutama open source -kirjastokin löytyy jo. Tässä työssä on käytetty OpenIEC61850 -kirjastoa.

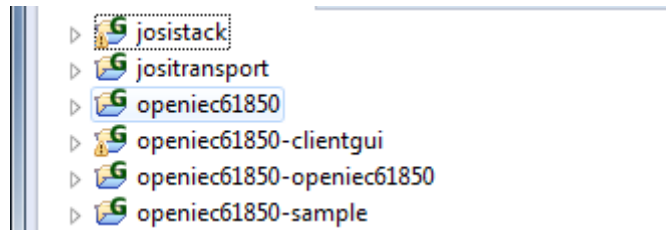
4.3 OpenIEC61850

OpenIEC61850 on OpenMUC -projektin luoma avoimen lähdekoodin IEC 61850 -kirjasto. Ohjelmointikielenä se käyttää Javaa. Se tarjoaa MMS clientin ja serverin IEC 61850:lle. ACSI:sta se tukee kaikkia GetDirectory ja GetDataDefinition -palveluita, GetDataValues ja SetDataValues -palveluita, Dataset -palveluita, puskuroimatonta raportointia ja control -palvelua. /1/

OpenIEC61850 koostuu viidestä eri komponentista. Nämä ovat:

1. openiec61850: kirjaston ydin.
2. josistack: ACSE -protokollaa implementoiva kirjaston osa, joka vastaa clientin ja serverin välisestä kommunikaatiosta. ACSE on kahden peer -entiteetin välisessä kommunikaatiossa käytetty protokolla. Josistack toimii yhdessä jositransportin (ks. alla) kanssa.
3. jositransport: josistackin kanssa clientin ja serverin kommunikaatiosta ja tiedonvälityksestä vastaava kirjaston osa. Implementoi RFC 1006 -tiedonsiirtopalvelua ja OSI -mallin kuljetuskerrosta.
4. openiec61850-sample: tarjoaa näyteclientin ja -serverin. Sisältää yksinkertaisen Javaohjelman, joka imitoi IEC 61850 -laitetta. Toinen osa ohjelmaa imitoi clientia.
5. openiec61850-clientgui: graafinen käyttöliittymä kirjastolle. Hakee laitteen datamallin serveriltä ja esittää sen graafisesti. Tukee read ja write -palveluja.

Näiden lisäksi kirjastoon kuuluu openiec61850 ja josistack -kirjastojen tarvitsema jasn1 -kirjasto enkoodamista ja dekoodaamista varten, slf4j-api ja logback-core/logback-classic loggaamiseen ja jcalendar ajan ilmaisemiseen client GUI:ssa. Kuvassa 13 kuvakaappaus Eclipseen tuoduista kirjaston komponenteista.



Kuva 13. OpenIEC61850 -kirjaston komponentit.

Sample -tiedostoja voidaan ajaa joko komentorivillä tai Eclipsessä. Komentorivillä ajettaessa mennään hakemistoon, jossa tiedostot sijaitsevat ja ajetaan joko runSampleServer.sh (Unixissa)/runSampleServer.bat (Windowsissa tiedosto pitää uudelleennimetä .bat:ksi ensin) tai runSampleClient.sh/runSampleClient.bat. Unixissa portti pitää myös olla alle 1000 jos ei omaa root -oikeuksia.

Kirjaston asentaminen Eclipseen koostuu seuraavista vaiheista:

1. Ladataan openiec61850 -kirjasto OpenMUC:n sivuilta (www.openmuc.org) ja puretaan se.
2. Asennetaan Eclipseen Gradle -lisäosa. Se on kirjastojen importoimiseen luotu työkalu, jolla voi tuoda Eclipseen laajoja projekteja. Gradle asennetaan menemällä Eclipsessä valikkoon Help → Eclipse Marketplace, laittamalla hakusanaksi Gradle ja asentamalla lisäosa Gradle Integration for Eclipse.
3. Gradlen asentamisen jälkeen Gradle -projekteja voidaan tuoda Eclipseen valitsemalla Import → Gradle project. Valitaan openiec61850, jossa on build.gradle -tiedosto luotuna valmiina nykyisissä versioissa, valitaan build model ja tuodaan projekti painamalla finish.
4. Lopuksi asetetaan kotikansio Gradlelle menemällä Eclipsessä Window →

Preferences → Java → Build Path → Classpath Variable ja luomalla muuttuja GRADLE_USER_HOME, jolle laitetaan poluksi käyttäjän kotikansiossa sijaitseva .gradle -kansio, jonka Gradlen asennus on sinne automaattisesti luonut.

Eclipsessä sampletiedostot ajetaan ajamalla SampleServer.launch ja SampleClient.launch -tiedostot (run as → sampleServer/sampleClient).

4.3 libiec61850

Libiec61850 on projekti, joka tarjoaa C-kielisen IEC 61850 -kirjaston. Projektin tarkoituksena on, että sitä voidaan ajaa sulautetuissa järjestelmissä ja mikro-ohjaimissa. Kirjasto myös tarjoaa yksinkertaisia esimerkkikoodeja, joita käyttäjä voi käyttää aloituspisteenä omille ohjelmilleen. Kirjasto pyörii Linux ja Windows -ympäristöissä. Kirjasto löytyy osoitteesta www.libiec61850.com. Se tukee MMS:ää, GOOSEa ja alustavasti myös SV:ä. /19/

Libiec61850:n olennaisimmat ominaisuudet on lueteltu alle:

- Täysi ISO:n protokollapino TCP/IP:n päällä
- IED:n staattinen mallinnus luomalla C -koodi SCL -tiedostosta
- Dynaaminen mallinnus API:lla
- Association -palvelu
- Read ja write -palvelut MMS -muuttujille
- Mallinselaamispalveluita (kuten GetServerDirectory ja GetDeviceDirectory, yms.)
- Dataset -palvelut, kuten datasettien luonti ja poisto
- IEC 61850 client ja server API
- MMS client

- Reporting -palvelu
- Tuki kaikille IEC 61850 control modeleille
- GOOSE -publisher ja -subscriber
- Hardware abstraction layer
- Työkalu SCL -tiedostojen muuntamiseksi IED -malleiksi tai server configuration -tiedostoiksi
- Tiedostojen ja kansioden selaaminen ja lukeminen
- Sopii myös rajalliset resurssit omaaville sulautetuille järjestelmille
- Tarjoaa myös ohjelmointirajapinnan C# clientille .NET -ympäristöön.

5 SOVELLUS

5.1 Yleistä

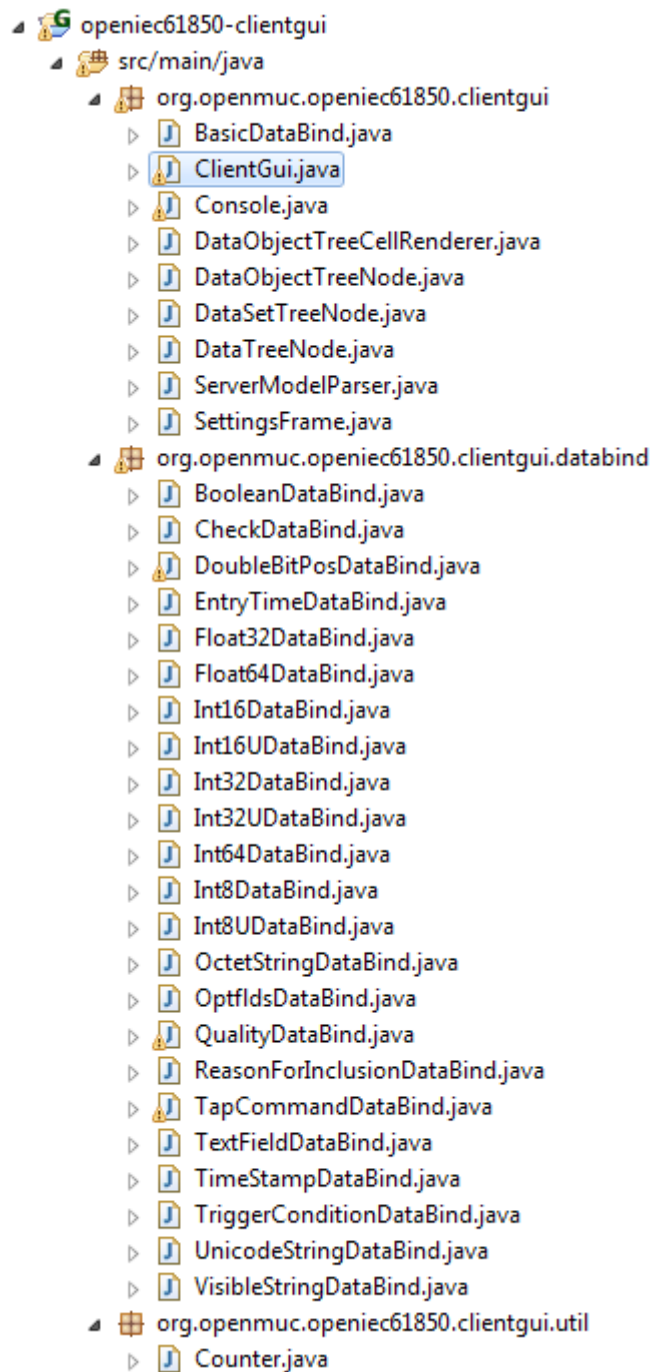
Tässä työssä standardin tutkimista varten on käytetty OpenIEC61850 -kirjastoprojektin Java -kielistä ohjelmistoa. Ohjelmalla voidaan visualisoida serverin tietorakenne. Ohjelmaan on myös lisätty konsoli loggausparametrejen seurausta varten sekä mahdollisuus visualisoida dataa SCL -tiedostosta.

Kirjaston clientgui -osassa, jossa graafinen toteutus luodaan on kolme pakkausta: org.openmuc.openiec61850.clientgui, org.openmuc.openiec61850.clientgui.databind ja org.openmuc.openiec61850.clientgui.util. Ensimmäisessä pakkauksessa ovat toiminnalliset luokat, jotka tekevät graafisen käyttöliittymän ohjelmalle. Pakkaukseen kuuluu:

- BasicDataBind.java: datansidonta -luokka
- ClientGui.java: graafisen käyttöliittymän luokka
- Console.java: konsolin luokka
- DataObjectTreeCellRenderer.java: renderöintiluokka
- DataObjectTreeNode.java: dataobjekteille tarkoitettu luokka
- DataSetTreeNode.java: dataseteille tarkoitettu luokka
- DataTreeNode.java: dataluokka, jonka avulla ilmaistaan dataa puumallisessa muodossa
- ServerModelParser.java: datanjäsennysluokka, pitää sisällään projektin olennaisimmat metodit
- SettingsFrame.java: asetuskehys ohjelmaan. Otettu pois käytöstä tässä työssä.

Toisessa pakkauksessa on datansidonta -luokat. Datansidonta -luokkia käytetään

yhteyden luomiseen käyttöliittymän ja datan välille. Standardin teoriassa vastine näille luokille on data-attribuuttien tyypit, kuten BdaInt8. Kolmas pakkaus on työkaluja varten luotu pakkaus, johon kuuluu yksi laskuriluokka: Counter.java. Kuvassa 14 on sovelluksen kaikki pakkaukset ja luokat.



Kuva 14. clientgui -projektin sisältö.

5.2 Toiminta

Kirjaston toiminnallisuuden keskeisimmät luokat kuuluvat sen openiec61850 -osioon. Tähän osioon kuuluvat esimerkiksi ClientAssociation, jota käytetään yhteysväylän luomiseen clientin ja serverin välille, ClientSap, joka edustaa yhteyspistettä IEC 61850 -clientien palveluille, ServerSap, joka edustaa yhteyspistettä IEC 61850 -serverien palveluille sekä standardin datamallin osia vastaavat luokat, kuten ConstructedDataAttribute, DataSet, Fc, FcDataObject, LogicalDevice ja LogicalNode.

Clientin ja serverin yhteys toimii projektissa siten, että ensin luodaan ClientSap -instanssi ohjelman client -osaan ja konfiguroidaan se. Koodin tasolla tämä suoritetaan kuvan 15 näyttämällä tavalla. Kommunikaatioon käytetään assosiaatio -palvelua, jota kutsutaan associate() -metodilla. Serveriä käytettäessä ensin haetaan lista palvelinten yhteyspisteistä käyttämällä ServerSap.getSapsFromScfFile() -metodia. Tämä metodi lukee SAP:it (Service Access Point, yhteyspiste) määrittelystä ICD -tiedostosta. Valitaan yhteyspiste, jota halutaan käyttää ja konfiguroidaan se. ServerSap:iin tuodaan laitteen täysi malli, joka on määritelty ICD -tiedostossa. Siitä voi saada kopion metodilla getModelCopy(). Kuuntelu aloitetaan startListening() -metodilla. Assosiaatiota kutsuttaessa määritetään serverin ip ja port sekä haluttaessa autentikaatioparametri. Autentikaatioparametri on laitteen tai serverin valmistajan tai ylläpitäjän määrittämä. Arvo voidaan jättää myös tyhjäksi. Kuvassa 15 kuvakaappaus kirjaston koodin kohdasta, jossa ClientSap luodaan.

```

ClientSap clientSap = new ClientSap(); //luodaan ClientSap instanssi

InetAddress address = null;
try {
    address = InetAddress.getByIp(ipTextField.getText()); //haetaan ip tekstikentältä
} catch (UnknownHostException e1) {
    e1.printStackTrace();
    return;
}

int remotePort = 10002;
try {
    remotePort = Integer.parseInt(portTextField.getText()); //haetaan portti tekstikentältä
    if (remotePort < 1 || remotePort > 0xFFFF) {
        throw new NumberFormatException("port must be in range [1, 65535]");
    }
} catch (NumberFormatException e) {
    e.printStackTrace();
    return;
}

```

Kuva 15. ClientSap -instanssin luonti.

IEC 61850 -server pitää sisällään puumaisen datamallin, jossa alimpina lehtinä on data (integerit, booleanit, stringit, jne.) joihin päästään käsiksi clientillä. Ylimpänä on server, jonka tyyppi OpenIEC61850:ssa on ServerModel. Serverin alla ovat loogiset laitteet, niiden alla loogiset nodet ja niiden alla dataobjektit. OpenIEC61850:ssa loogiset nodet eivät sisällä täysiä dataobjekteja vaan ne sisältävät functional constraint -dataobjekteja. Kaikki nodet OpenIEC61850:n server -mallissa ovat yhtä seitsemästä tyypistä: ServerModel, LogicalDevice, LogicalNode, FcDataObject, Array, ConstructedDataAttribute ja BasicDataAttribute. Kaikki nämä implementoivat ModelNode -rajapintaa. Nodet FcDataObject, Array, ConstructedDataAttribute ja BasicDataAttribute myös implementoivat FcModelNode -rajapintaa, koska ne on määritelty standardissa functional constraint -dataksi. Monia IEC 61850:n palveluista voidaan käyttää ainoastaan functional constraint -dataan (kuten GetDataValues ja SetDataValues).

Clientiä ohjelmoidessa serveristä saadaan kopio joko ClientAssociation.retrieveModel() -metodilla tai ClientAssociation.getModelFromScfFile() -metodilla. Serveriä ohjelmoitaessa serveristä saadaan kopio ServerSap.getModelCopy() -toiminnolla. Kun malli on haettu, sitä voidaan navigoida muutamalla toiminnolla. Ensimmäinen näistä on

ServerModel.findModelNode(ObjectReference objectReference, Fc fc), joka etsii alinoden annetulla referenssillä ja functional constraintilla. Toinen metodi, jota voidaan käyttää on ModelNode.getChild(String name, Fc fc), joka palauttaa alinoden annetulla nimellä ja functional constraintilla. Viimeisenä on ModelNode.getBasicDataAttributes(), joka palauttaa listan model noden kaikista alinodeista. ClientAssociationin alta löytyy myös muita hyödyllisiä metodeja, kuten createDataSet (datasettien luomiseen käytetty metodi), deleteDataSet (datasetin poistamiseen käytetty metodi), enableReporting (raportoinnin päälle kytkeminen), getReport (pyydetään raportti), operate (ACSI:n operate -palvelun vastine) ja select (ACSI:n select -metodin vastine). Kuvassa 16 kuvakaappaus kirjaston koodista, missä luodaan assosiaatio sekä haetaan palvelimen data serverModel -muuttujaan.

```

try {
    association = clientSap.associate(address, remotePort, null); //luodaan assosisaatio
} catch (IOException e) { //ClientSap:n ja ServerSap:n välille
    logger.error("Error connecting to server: " + e.getMessage());
    return;
}

ServerModel serverModel;

try {
    serverModel = association.retrieveModel(); //metodi kutsuu kaikki GetDirectory
    association.getAllDataValues(); //!!! //ja GetDefinition ACSI -palvelut serverin mallin hakemiseen
} catch (ServiceError e) {
    logger.error("Service Error requesting model.", e);
    association.close();
    return;
} catch (IOException e) {
    logger.error("Fatal IOException requesting model.", e);
    return;
}

```

Kuva 16. Serverin datamallin noutaminen käyttämällä assosiaatiota.

Ohjelmalla on neljä metodia toimintoja varten: connect(), reload(), write() ja fromscl(). Connect() -metodia käytetään yhteyden luomiseen. Siinä luodaan ServerModel -luokan muuttuja ja haetaan serverin looginen malli association -muuttujaan retrieveModel() -metodia käyttäen. Reload() lataa näytettyjen nodejen arvot uudelleen. Tähän käytetään reset -komentoa. Write() kirjoittaa noden uudelleen, mikäli se on kirjoitettavissa. Fromscl() -metodilla haetaan data

erillisestä SCL -tiedostosta. Kuvassa 17 näkyy, kuinka fromscl() -metodia käytetään.

```
private void fromscl() {  
  
    JFileChooser fc = new JFileChooser(); //Javan filedialogue  
    fc.showOpenDialog(readFromSclFile);  
    String filePath = fc.getSelectedFile().getAbsolutePath();  
  
    ServerModel serverModel;  
  
    try {  
  
        serverModel = association.getModelFromSclFile(filePath); //haetaan määritetystä tiedostosijainnista  
    } catch (SclParseException e1) { //sijaitsevasta SCL -tiedostosta datamalli  
        logger.error("Error parsing SCL file.", e1);  
        return;  
    }  
}
```

Kuva 17. Datamallin nouto SCL -tiedostosta.

Ohjelman graafinen esitys luodaan pääasiassa ServerModelParser ja ClientGui -luokilla sekä näiden apuluokilla. ServerModelParser on se luokka, missä assosiaatiolla haettu data jäsennetään graafiseen esitykseen sopivaksi. Kuvassa 18 näkyy, kuinka ServerModelParseria käytetään datan jäsentämiseen. Alla on luokan metodit ja niiden toimintatarkoitus ja toiminta selvitettyinä:

1. createModelTree(): metodi luo puumallin, johon looginen laite ja loogiset nodet tallennetaan. Tallentuu DataObjectTreeNode -muodossa. Siinä myös kutsutaan addLogicalDevice() -metodia sekä DataSettien luomismetodia.
2. addLogicalDevice(): loogisen laitteen luomismetodi. Metodi luo loogisen laitteen, LLNO:n, ja tallentaa sen DataObjectTreeNodeksi. Kutsuu addLogicalNode() -metodia.
3. addLogicalNode(): loogisten nodejen luomismetodi. Metodi hakee kaikki loogisen laitteen loogiset nodet ja tallentaa ne HashMap:iin. Kutsuu myös addFunctionalConstraintObject -metodia. Käyttää DataObjectTreeNode -luokkaa tallennusmuotona. Kuvassa 19 on metodin koodi.
4. addDataSet(): mahdollisten DataSettien luomismetodi. Tallennetaan

DataSetTreeNode -muotoon.

5. addFunctionalConstraintObject(): luo functional constraint -objektin, FCDO:n. Tallennetaan DataObjectTreeNodena.
6. addDataObject(): dataobjektien luomismetodi. Tallennus DataObjectTreeNodena.

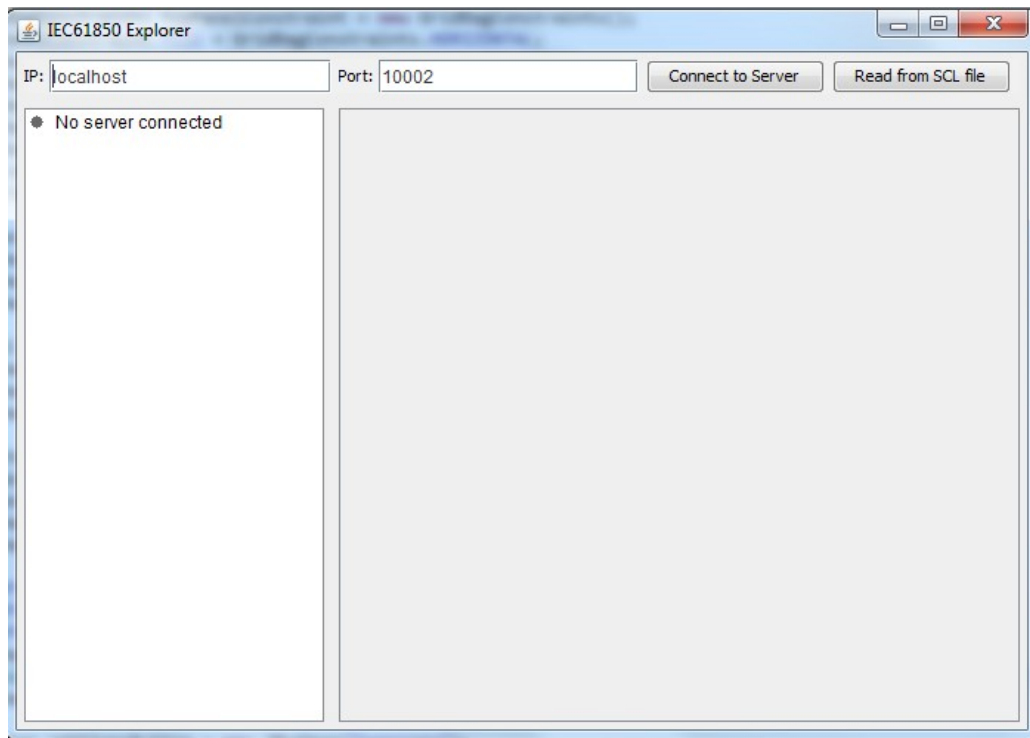
```
ServerModelParser parser = new ServerModelParser(serverModel); //jäsenetään serverModel serverModelParserin
tree.setModel(new DefaultTreeModel(parser.getModelTree())); //avulla graafiseen käyttöliittymään sopivaksi
```

Kuva 18. serverModelin jäsentäminen.

```
private void addLogicalNode(DataObjectTreeNode parent, LogicalNode node) {
    DataObjectTreeNode treeLN = new DataObjectTreeNode(node.getName(), node);
    parent.add(treeLN);
    Collection<ModelNode> children = node.getChildren();
    Map<String, Set<Fc>> childMap = new HashMap<String, Set<Fc>>();
    for (ModelNode child : children) {
        if (!childMap.containsKey(child.getName())) {
            childMap.put(child.getName(), new HashSet<Fc>());
        }
        childMap.get(child.getName()).add(((FcModelNode) child).getFc());
    }
    for (Entry<String, Set<Fc>> childEntry : childMap.entrySet()) {
        addFunctionalConstraintObject(treeLN, node, childEntry.getKey(), childEntry.getValue());
    }
}
```

Kuva 19. addLogicalNode() -metodi.

ClientGui -luokka hoitaa graafisen käyttöliittymän luonnin. Luokka käyttää Eclipsen sisäänrakennettuja grafiikkakomponentteja ulkoasun luomiseen. Luokassa on kaksi tekstikenttää, ipTextField ja portTextField, serverin IP:ä ja porttia varten, nappi yhdistämisfunktion kutsua varten sekä nappi SCL -tiedostosta lukemisen funktion kutsumista varten. Ohjelman Layout -muotona on GridBagLayout. Kuvassa 20 kuva ohjelman käyttöliittymästä.



Kuva 20. Ohjelman graafinen käyttöliittymä.

ClientGuin ja ServerModelParserin lisäksi ohjelman clientgui -pakkaukseen kuuluu jo edellä ServerModelParserin metodien selvityksessä esille tulleet DataObjectTreeNode ja DataSetTreeNode. Näitä luokkia käytetään serverin datamallin visualisoinnissa ClientGuin ja ServerModelParserin kanssa. DataObjectTreeNodea käytetään kaikkien dataobjekteiksi luokiteltavien tietojen visualisointiin ja DataSetTreeNodea käytetään datasettien visualisointiin.

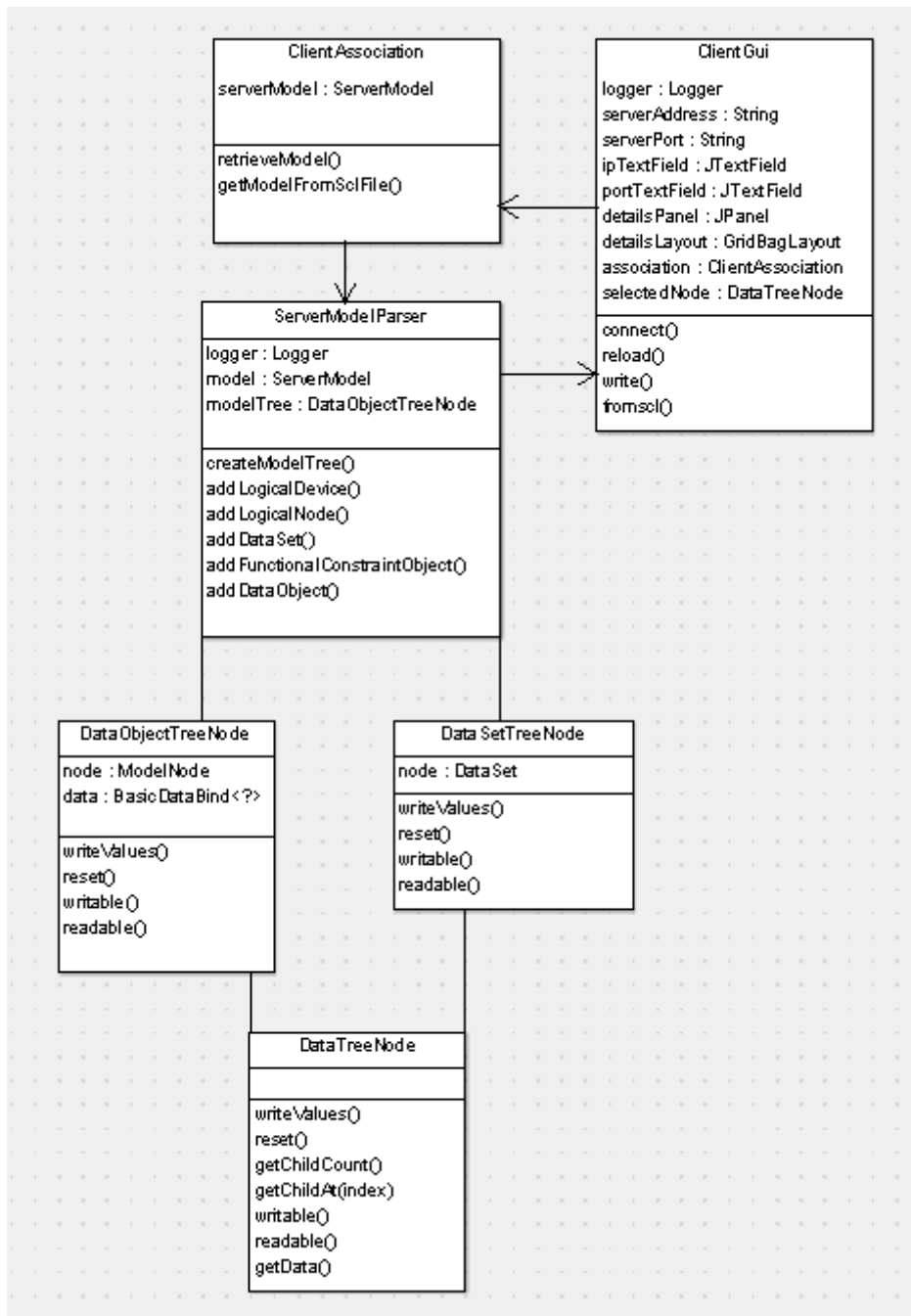
Molemmat edellä mainituista luokista implementoivat DataTreeNode -luokkaa, lyhyttä luokkaa jossa on muutama metodi datan noutoa varten. Nämä metodit ovat:

1. `getData()`: hakee objektin datan.
2. `reset()`: lataa arvon uudelleen
3. `writeValues()`: kirjoittaa arvon
4. `getChildCount()`: hakee objektin osien määrän. Esim. hakee dataobjektin data-attribuuttien määrän.

5. getChildAt(int index): hakee objektin osan määritetystä sijainnista. Parametrinä käytetään int -lukua.
6. writable(): selvittää onko objekti kirjoitettavissa uudelleen. Palauttaa tiedon boolean -muodossa.
7. readable(): selvittää onko objekti luettavissa. Palauttaa tiedon boolean -muodossa.

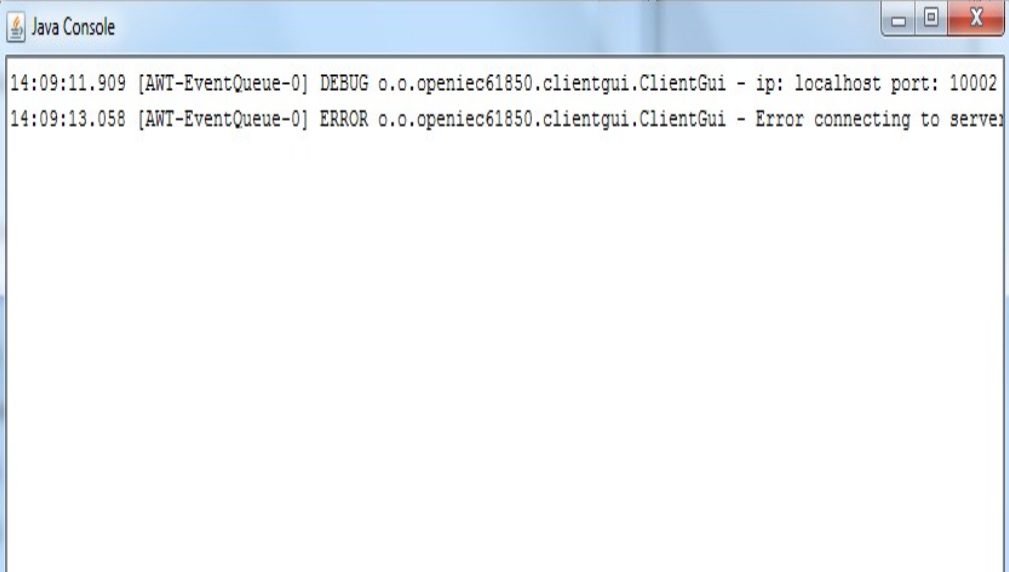
Sen lisäksi, että DataObjectTreeNode ja DataSetTree -luokat implementoivat DataTreeNode -luokkaa, ne jatkavat javax.swing -kirjaston DefaultMutableTreeNode -luokkaa.

Ohjelman toiminnan logiikka on kuvattuna kuvaan 21.



Kuva 21. Luokkakaavio ohjelman toimintaperiaatteesta.

Lisäksi ohjelmaan lisättiin konsoli, joka pitää logikirjaa tapahtumista ja virheilmoituksista. Tapahtumat kirjataan ylös Logger -olioon, ja Console -luokka tulostaa ne. Kuvassa 22 on konsolin ulkoasu.



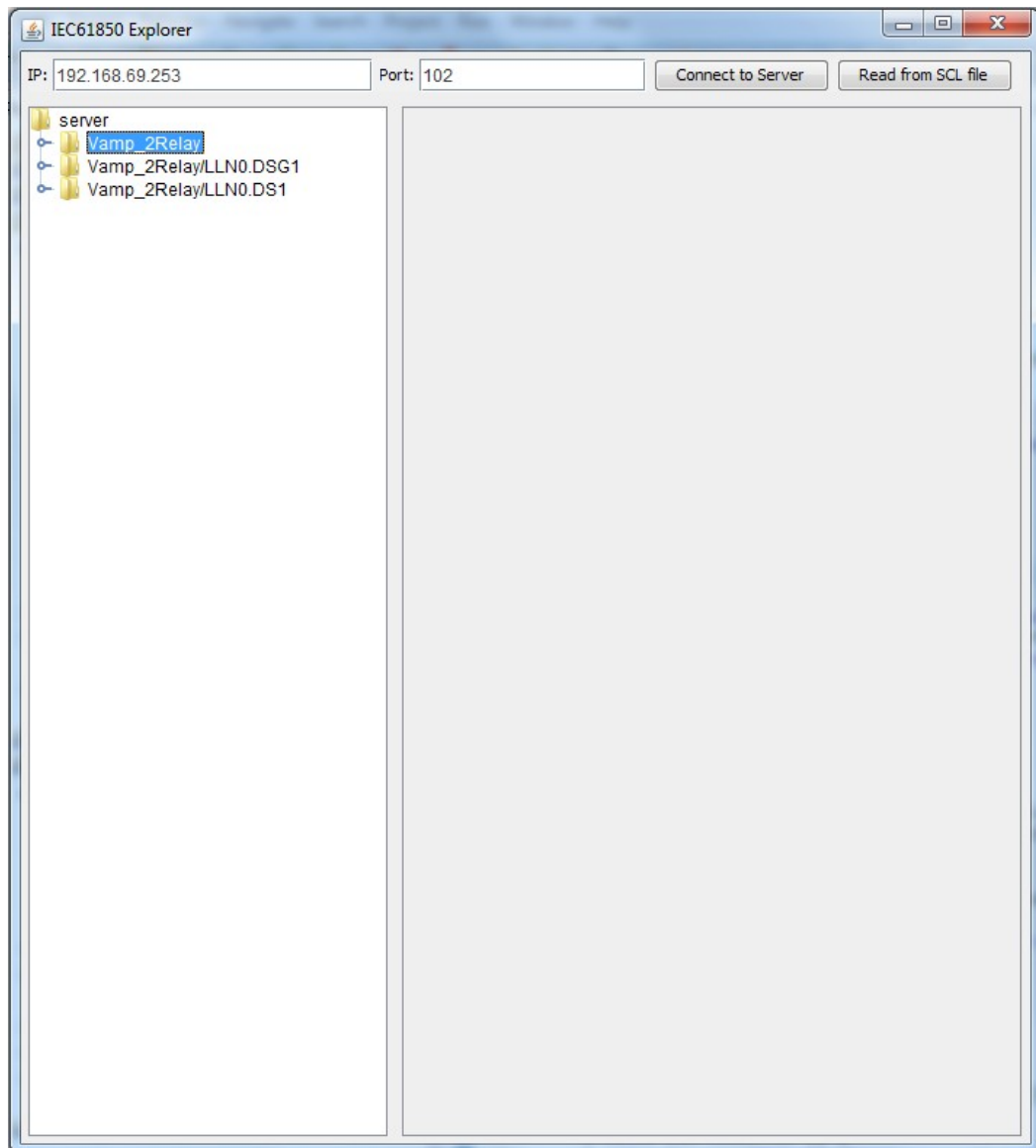
```
Java Console
14:09:11.909 [AWT-EventQueue-0] DEBUG o.o.openiec61850.clientgui.ClientGui - ip: localhost port: 10002
14:09:13.058 [AWT-EventQueue-0] ERROR o.o.openiec61850.clientgui.ClientGui - Error connecting to server
```

Kuva 22. Ohjelman konsoli.

6 TULOKSET

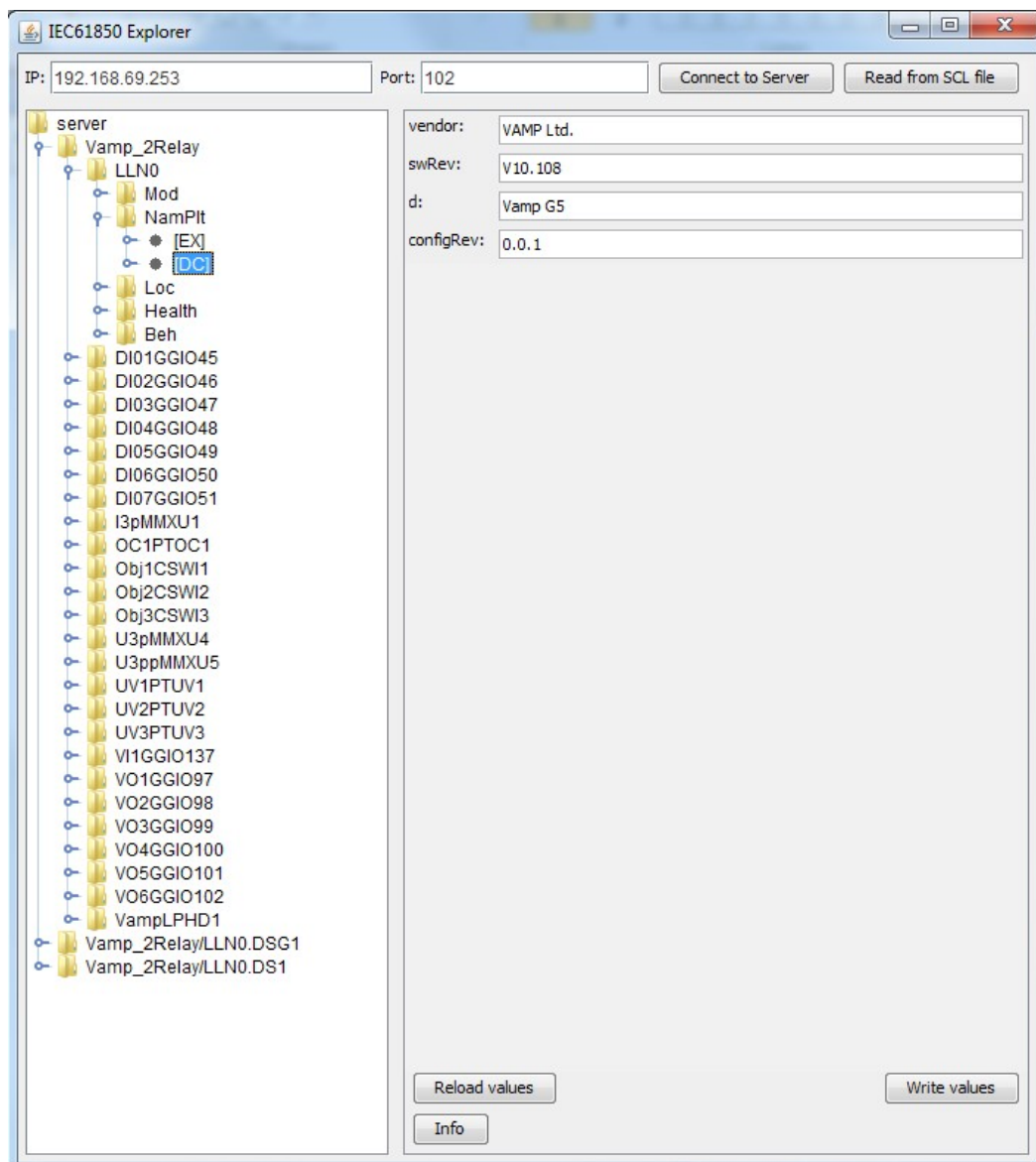
6.1 Datan nouto palvelimelta

Kuvassa 23 on kuvakaappaus siitä, minkälainen näkymä aukeaa kun ohjelmalla luodaan yhteys Vampin 257 -suojareleeseen. Releestä löytyy 3 datasettiä: Vamp_2Relay, Vamp_2Relay/LLN0.DSG1 ja Vamp_2Relay/LLN0.DS1.



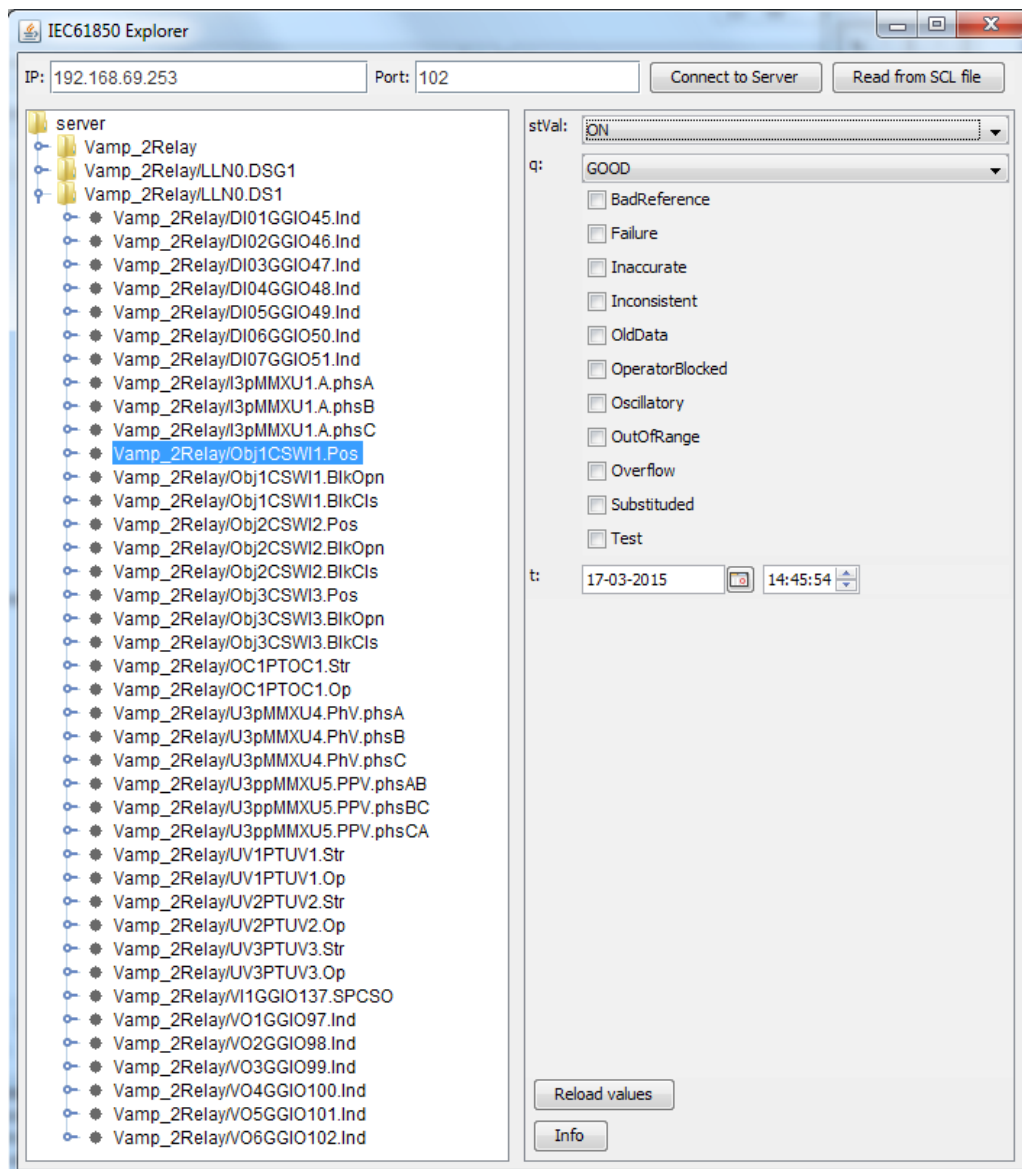
Kuva 23. Vamp 257- suojarleen SCL- tiedosto.

Ylimmästä datasetistä, Vamp_2Relay -datasetistä löytyy datasettejä, jotka kuvaavat releen fyysisiä ominaisuuksia. Kuvassa 24 tämän datasetin sisältö.



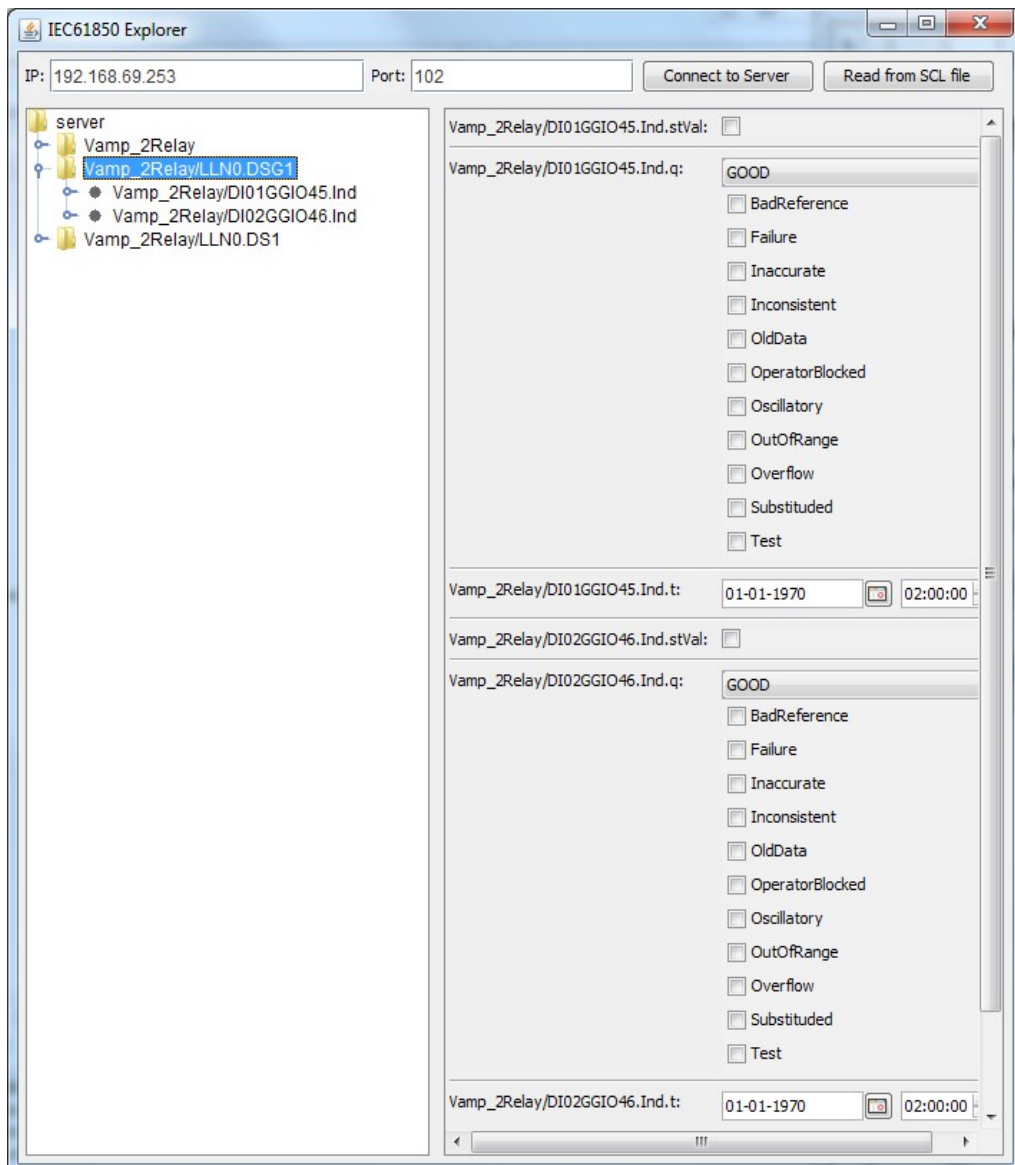
Kuva 24. Vamp 257 -releen data ja tunnistuskilven kuvaustieto.

Vamp_2Relay/LLN0.DS1:n sisältö on kuvattuna kuvaan 25. Datasetsien sijaan se sisältää listan yleisesti käytetyistä nodeista. Kuvassa valittuna CSWI.Pos -dataobjekti.



Kuva 25. Vamp 257 -releen data ja SWI -noden Pos -data.

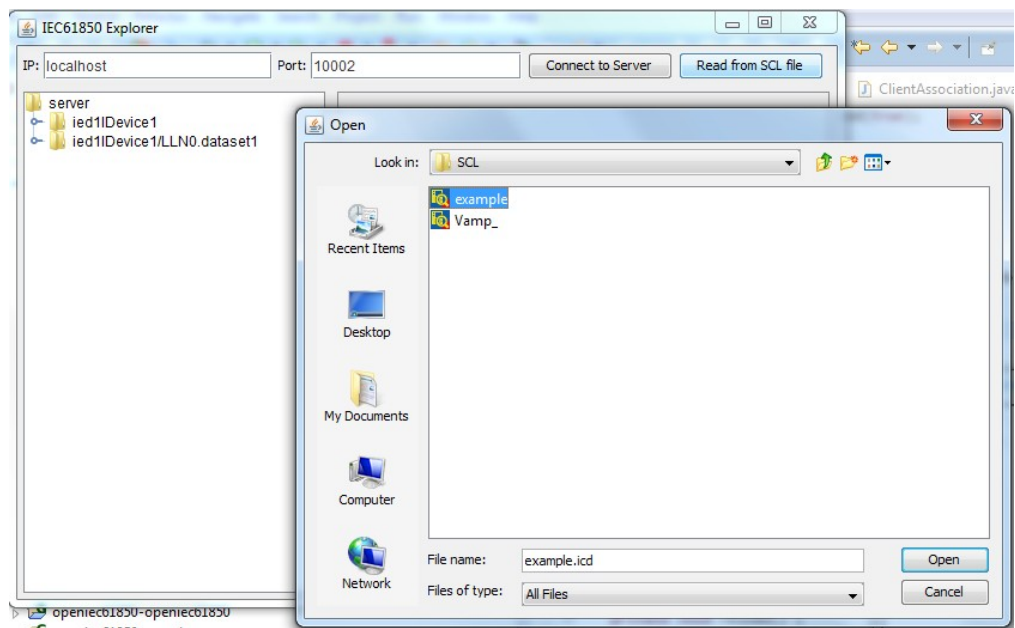
Viimeisessä datasetissä, Vamp_2Relay/LLN0.DSG1:stä löytyy kaksi objekta, jotka liittyvät digitaaliseen ulostuloon. Kuvassa 26 kuva näistä objekteista.



Kuva 26. LLN0.DSG1 -datasetin data.

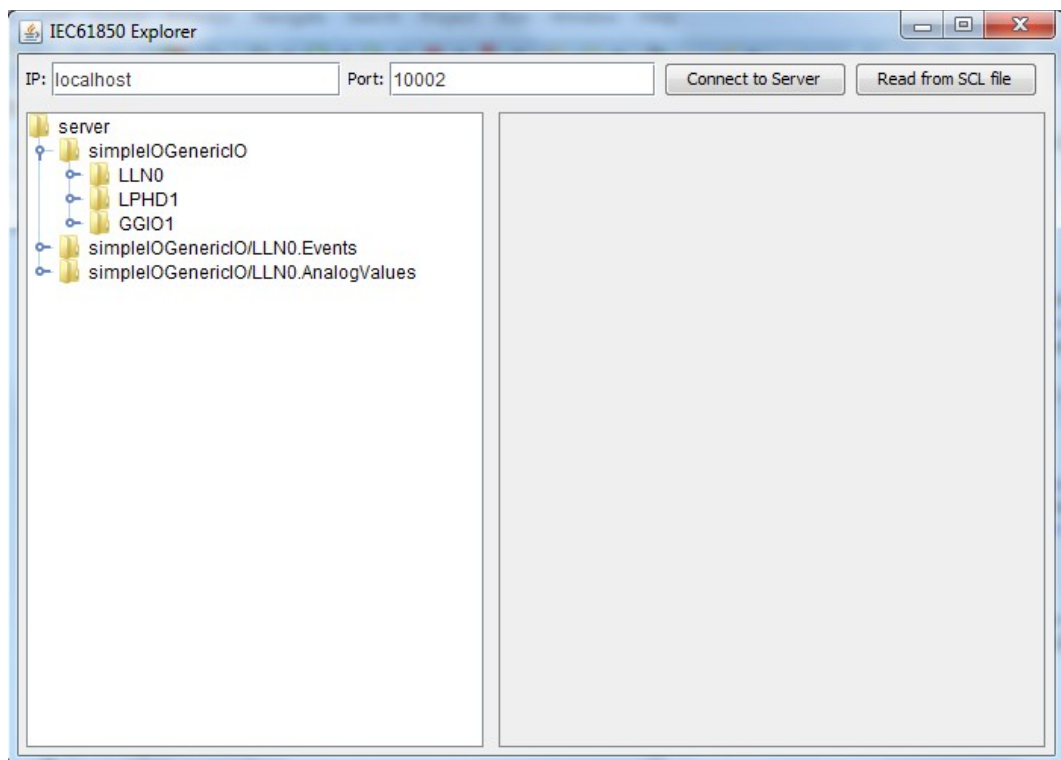
6.2 Datan nouto SCL -tiedostosta

Kuvassa 27 on kuva ohjelman FileDialogesta. Esimerkkinä on käytetty example.icd -tiedostoa, jonka malli on esitelty tässä työssä SCL:ää koskevassa luvussa. Tiedoston data näkyy kuvassa 28.



Kuva 27. Tiedoston avausdialogi.

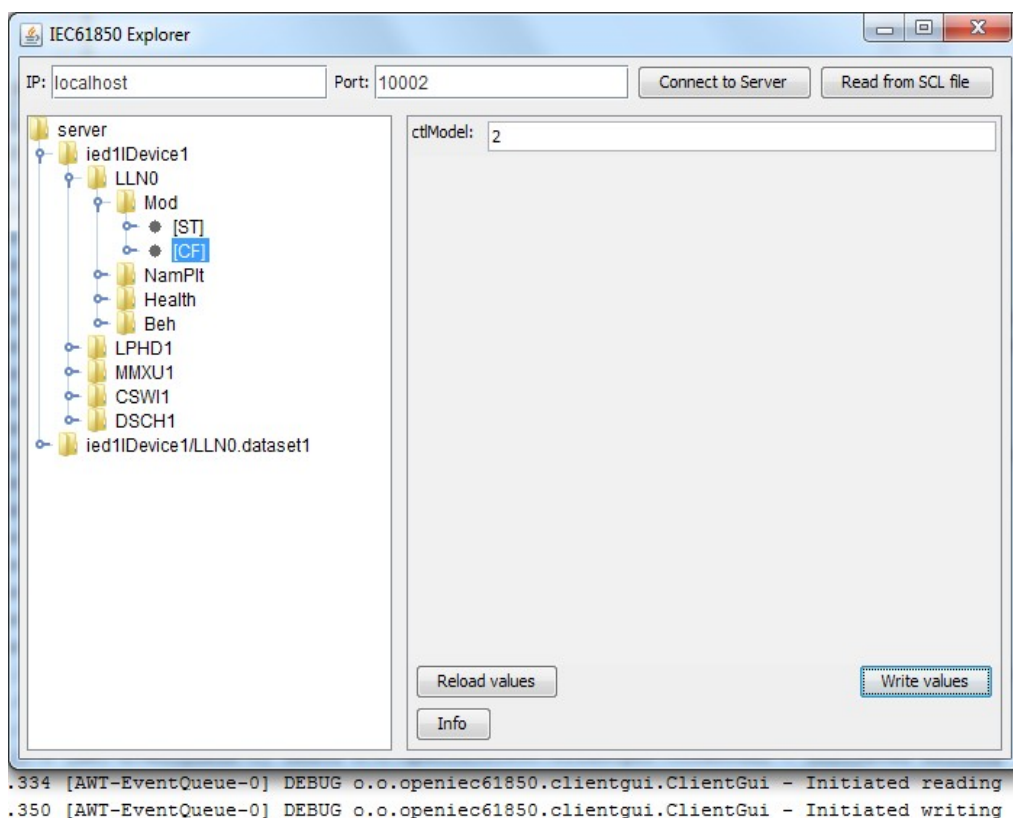
Tiedostossa on kolme datasettiä: datasetti fyysiselle datalle (LLN0, LPHD ja GGIO), LLN0 tapahtumille ja LLN0 analogiselle datalle.



Kuva 28. Example.icd:n data.

6.3 Write -funktio

Ohjelman write -funktioita on testattu kirjoittamalla OpenIEC61850 SampleServerin LLN0:n Mod -objektista, CF -functional constraintin data ctlModel (control model) arvo uusiksi. Kuvassa 29 kuvattuna arvon kirjoittaminen. Alkuarvona ollut 0, uutena arvona 2. Clientin puolella saadaan loggeriin viesti 'Initiated writing'.



Kuva 29. Write -funktion käyttö SampleServerillä.

Serverin puolella saadaan ilmoitus kirjoituspyynnön vastaanottamisesta. Viesti kuvattuna kuvaan 30.

```
12:43:08.351 [pool-1-thread-2] DEBUG o.o.openiec61850.ServerAssociation - Got a Write request
12:43:08.351 [pool-1-thread-2] DEBUG o.o.openiec61850.ServerAssociation - Got a SetDataValues request.
12:43:08.351 [pool-1-thread-2] INFO o.o.openiec61850.sample.SampleServer - got a write request: ied1Device1/LLN0.Mod.ctlModel: 2
```

Kuva 30. Kirjoituspyynnön vastaanottaminen.

7 JOHTOPÄÄTÖKSET

Työssä tutkittiin IEC 61850 -standardia sekä sitä hyödyntävää OpenIEC61850 -kirjastoa. Työssä käydään läpi standardin teorian avainosat: datamalli, ACSI, SCSM ja SCL. Datamalliosiossa selitetään kuinka sähköaseman toiminnot jakautuvat standardissa osiin. ACSI -osiossa selitetään kuinka dataa käsitellään standardissa. SCSM -osiossa selitetään standardin tiedonsiirtotavat. SCL -osiossa selitetään standardin konfiguraatiokieli.

OpenIEC61850 -kirjastosta selitetään kirjaston eri osat ja kuinka se toimii. Työssä tuotettiin myös kirjastoa hyödyntäen ohjelmisto, jolla voidaan visualisoida IEC 61850 -palvelimen tai SCL -tiedoston dataa sekä kirjoittaa dataa palvelimelle. Palvelimelta visualisoitaessa kutsutaan standardin kaikki GetDefinition ja GetDirectory -palvelut, jotka sen jälkeen jäsennetään ServerModelParser -luokalla graafiseen käyttöliittymään esitettäväksi. SCL -tiedostosta noudettaessa tehdään sama, mutta käytetään siihen luotua erityistä metodia. Dataa kirjoitettaessa kutsutaan SetDataValues -palvelua.

Kaiken kaikkiaan, vaikka työssä on tuotettu suhteellisen vähän esimerkiksi itsetehtyä koodia, on se kuitenkin tarjonnut hyvin paljon tekemistä. Ennen kuin on pystytty aloittamaan itse standardin opettelu on täytynyt opetella sähköasemien toiminta, ja ennen kuin on voitu aloittaa kirjaston käytön opettelu on täytynyt opetella standardi. Työtä varten on jouduttu lukemaan kaksi kirjaa, useita artikkeleita sekä standardin määrittelytiedostot. Erityisesti viimeiseksi mainitut vaativat paljon ajattelua, jotta niistä saa muodustettua kuvan itselleen.

Lopputuloksena on tutkimustyö, jossa standardin kokonaisuus on tiivistetty insinööriopiskelijan tasoiseksi tekstiksi. Työn OpenIEC61850 -osia voidaan käyttää kirjaston toimintaperiaatteen sisäistämiseen, josta voidaan siirtyä ohjelmoimaan monimutkaisempia ohjelmistoja. Kirjastoa kehitetään koko ajan ja sen perustuessa avoimeen lähdekoodiin kuka vain voi kehittää sitä edelleen. Tässä työssä esitellyt ominaisuudet ovat vain perusominaisuuksia, joista voidaan siirtyä pidemmälle. Työstä tehty wikisivusto tiivistää aihetta edelleen. Wiki löytyy osoitteesta <http://omega.cc.puv.fi/redmine/projects/edu-iec61850/wiki>.

LÄHTEET

- /1/ Fraunhofer Institute for Solar Energy Systems 2015. OpenIEC61850 – Overview. Viitattu 15.4.2015. <http://openmuc.org/index.php?id=35>
- /2/ Gunther, E. Features and Benefits of IEC 61850. Viitattu 17.3.2015.
- /3/ Harsia, P. 2013. Kiinteistön sähköjakelu. Viitattu 9.3.2015. <http://tate.blogs.tamk.fi/sahkoinen-talotekniikka/sahkoverkko/kiinteiston-sahkoverkko/>
- /4/ International Electrotechnical Commission 2003. IEC61850-1 Introduction and overview. Viitattu 17.3.2015.
- /5/ International Electrotechnical Commission 2003. IEC61850-2 Glossary. Viitattu 17.3.2015.
- /6/ International Electrotechnical Commission 2003. IEC61850-3 General requirements. Viitattu 17.3.2015.
- /7/ International Electrotechnical Commission 2010. IEC61850-4 System and project management. Viitattu 17.3.2015.
- /8/ International Electrotechnical Commission 2003. IEC61850-5 Communication requirements for functions and device models. Viitattu 17.3.2015.
- /9/ International Electrotechnical Commission 2010. IEC61850-6 Configuration language for communication in electrical substations related to IEDs. Viitattu 14.4.2015.
- /10/ International Electrotechnical Commission 2010. IEC61850-7-1 Principles and models. Viitattu 17.3.2015.
- /11/ International Electrotechnical Commission 2010. IEC61850-7-2 Abstract communication service interface (ACSI). Viitattu 14.4.2015.
- /12/ International Electrotechnical Commission 2010. IEC61850-7-3 Common Data Classes. Viitattu viitattu 17.3.2015.
- /13/ International Electrotechnical Commission 2010. IEC61850-7-4 Compatible logical node classes and data classes. Viitattu 17.3.2015.
- /14/ International Electrotechnical Commission 2010. IEC61850-8-1 Mappings to MMS (ISO/IEC9506-1 and ISO/IEC 9506-2). Viitattu 17.3.2015.
- /15/ International Electrotechnical Commission 2010. IEC61850-9-2 Sampled values over ISO/IEC 8802-3. Viitattu 17.3.2015.

/16/ International Electrotechnical Commission 2010. IEC61850-10
Conformance testing. Viitattu 17.3.2015.

/17/ Starck, J. 2010. IEC 61850. Viitattu 16.4.2015.

/18/ Vaasan ammattikorkeakoulu 2014. DEMVE training material. Viitattu
16.4.2015.

/19/ Zillgith, M. 2015. About. Viitattu 15.4.2015.
<http://libiec61850.com/libiec61850/about/>