



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MAURI MUSTONEN
SÄHKÖASEMAN ÄLYKKÄÄN ELEKTRONIIKKALAITTEEN
VIESTIEN TILAUS JA PROSESSOINTI
Diplomityö

Tarkastaja: Professori Kari Systä

Jätetty tarkastettavaksi 17. touko-
kuuta 2018

TIIVISTELMÄ

MAURI MUSTONEN: sähköaseman älykkään elektroniikkalaitteen viestien tilaus ja prosessointi
 Tampereen teknillinen yliopisto
 Diplomityö, 20 sivua
 Toukokuu 2018
 Tietotekniikan koulutusohjelma
 Pääaine: Ohjelmistotuotanto
 Tarkastaja: Professori Kari Systä

Avainsanat: Ohjelmistotuotanto, IEC 61850, MMS, AMQP

Tiivistelmä on suppea, 1 sivun mittainen itsenäinen esitys työstä: mikä oli ongelma, mitä tehtiin ja mitä saatiin tulokseksi. Kuvia, kaavioita ja taulukoita ei käytetä tiivistelmässä.

Laita työn pääkielellä kirjoitettu tiivistelmä ensin ja käänös sen jälkeen. Suomenkielille kandidaatintyölle pitää olla myös englanninkielinen nimi arkistointia varten.

Sähkönjakeluverkko on tärkeä osa nykyistä yhteiskuntaa ja sen päivittäistä toimintaa. Siksi verkon toiminta on elintärkeää yhteiskunnan sujuvan toiminnan kannalta. Sähköverkko koostuu sähköntuotantolaitoksista, sähkölinjoista ja sähköasemista. Sähköverkon eri komponenttien avulla sähkö toimitetaan tuotantolaitoksesta kuluttajan seinäpistokeeseen asti. Sähköasemat ja niiden automatisointi ovat tärkeässä roolissa verkon yleisen toiminnan ja turvallisuuden takaamiseksi.

Nykypäivänä sähköverkon sähköasemien automatisointiin käytetään tietokoneita, kommunikointi- ja verkkolaitteistoja [11, s. 659]. Asemien automatisointi toteutetaan *älykkäillä elektroniisilla laitteilla* (eng. IED), jotka toteuttavat aseman monitoroinnin ja ohjaamisen. IED:t konfiguroidaan toteuttamaan aseman erilaisia funktioita ja toiminnallisuuksia. Monen eri toimijan toimiessa allalla, voisivat kaikki vapaasti toteuttaa oman version laitteistaan ja määrittää kuinka näitä käytettäisiin. Tästä seurauksena olisi, että laitteet eivät olisi yhteensopivia muiden toimittajien laitteiden kanssa. Tilannetta helpottamaan ja mahdollistamaan eri valmistajien yhteensopivuuden laitteiden välillä. On *International Electrotechnical Commission* määrittänyt standardin nimeltä IEC 61850. Standardi määrittää kommunikointi protokollat sähköasemien IED laitteille. Standardi määrittää abstrakteja malleja sähköaseman kommunikoinnin mallintamiseen oliopohjaisesti. Tekniikasta tai toteutuksesta riippumaton standardimalli voidaan määrittää toimivaksi eri tekniikoilla. Standardi määrittää malleja myös erilaisten datapisteiden tilaukseen viesteinä IED:ltä. Eri asiakastoteutukset voivat tilata laitteeseen määritettyjä datapisteitä standardin määrittämillä säännöillä. Standardin ansiosta viestien muoto on ennaltamäärätty ja on tekniikasta riippumaton. Asiakasohjelmisto voi käyttää viestejä esimerkiksi jännitemittatietojen tilaukseen.

Tässä diplomityössä keskitytään yllämainitun viestejä tilaavan asiakasohjelmiston suunnitteluun ja toteutukseen. Työ sisältää tutkimusta ja vertailua arkkitehtuuriin ja toteutukseen liittyen. Lopullinen toteutus asiakasohjelmistosta tilaa viestejä IED:ltä, prosessoi ja uudelleenjulkaisee ne myöhempää käyttöä varten. Ennen työn aloitusta, ohjelmistosta oli jo saatavilla protoversio, jota hyödynnettiin uuden toteutuksen suunnittelussa. Protoversiossa olevat ongelmat ja puutteet korjattiin uuteen toteutukseen.

ALKUSANAT

Mistä tämän diplomityönaiheen sain ja kiittää eri ihmisiä ketä työssä oli sidoshenkilöinä.

Tampereella, 19.4.2018

Mauri Mustonen

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Tausta	2
1.2	Laajuus	2
1.3	Tavoitteet.....	2
1.4	Työn rakenne.....	3
2.	TEORIA.....	4
2.1	IEC 61850 -standardi	4
2.1.1	Standardin historia	5
2.1.2	Standardin eri osat	5
2.1.3	Standardin määrittämä abstraktimalli	6
2.1.4	Viestiblokin konfigurointi ja tilaus	7
2.1.5	Viestin rakenne	7
2.2	Abstraktimallin sovitus MMS-protokollaan	7
2.2.1	MMS-protokolla	7
2.3	Advanced Message Queuing Protocol	7
2.3.1	Viestien välitysmekanismit.....	7
2.3.2	Tilaus ja julkaisu -mallin osat.....	7
3.	ALKUTILANNE	8
3.1	Kokonaiskuva.....	9
3.2	Ratkaistavat ongelmat	9
3.3	Tutkimuskysymykset	10
4.	SUUNNITTELU	11
4.1	Suorituskyvyn parantaminen	11
4.2	Järjestelmän hajautus	11
4.3	Ohjelmiston parametrisointi.....	11
4.4	Arkkitehtuurin suunnittelu	11
4.5	Prosessoidun viestin muoto.....	11
5.	TOTEUTUS	12
5.1	Ohjelmiston toteutuksen valinta.....	12
5.2	Kielen valinta	12
5.3	RabbitMQ	12
5.4	Käytettävät kirjastot	12
5.4.1	libiec61850	12
5.4.2	rabbitmq-c.....	13
5.4.3	JSON-formatointi	13
5.5	Jatkokehitys.....	13
6.	ARVIOINTI	14
7.	TULOKSET	15

8. YHTEENVETO.....	16
LÄHTEET.....	19

KUVALUETTELO

<i>Kuva 1.</i>	<i>IEC 61850 -standardin osat ja niiden väliset relaatiot [3, s. 14].</i>	<i>.....</i>	6
-----------------------	---	--------------	----------

TAULUKKOLUETTELO

<i>Taulukko 1. IEC 61850 -standardin pääkohtien ja niiden alakohtien dokumentit. ...</i>	5
--	---

LYHENTEET JA MERKINNÄT

ACSI	engl. <i>Abstract Communication Service Interface</i> , IEC 61850 -standardin käyttämä lyhenne kuvaamaan palveluiden abstraktimalleja
AMQP	engl. <i>Advanced Message Queuing Protocol</i>
FFI	engl. <i>Foreign Function Interface</i> , mekanismi, jolla ajettava ohjelma voi kutsua toisella kielellä implementoitua funktiota
GIL	engl. <i>Global Interpreter Lock</i> , tulkattavassa kielissä oleva globaali lukitus, joka rajoittaa yhden säikeen suoritukseen kerrallaan
HAL	engl. <i>Hardware Abstraction Layer</i> , laitteistoabstraktiotaso abstraktoimaan laitteen toiminnallisuus lähdekoodista
IED	engl. <i>Intelligent Electronic Device</i> , sähköaseman älykäs elektroninen laite, joka tarjoaa toimintoja monitorointiin ja kontrollointiin
MMS	engl. <i>Manufacturing Message Specification</i>
RCB	engl. <i>Report Control Block</i> , raporttien konfigurointiin ja tilaukseen tarkoitettu lohko asiakasohjelmalle
XML	engl. <i>Extensible Markup Language</i> , laajennettava merkintäkieli, joka on ihmis- ja koneluettava

1. JOHDANTO

Kirjoita tähän johdantoa työstä ja aiheesta. Kuinka työ valittiin ja miksi tekijä valitsi tämän työn. Kirjoita myös mitä tehtiin. Kokonaiskuva työstä pitäisi saada johdannosta. Alusta lukijaa todella hyvin yleismaallisella kuvalla ja taustalla. Asiaa pitäisi olla hyvin hallussa ennen teoriaosuuteen siirtymistä.

Tämän osion lukemalla lukijan pitää tietää:

- *Miksi lukija valitsi tämän aiheen?*
- *Mikä on sähköaseman IED ja mitä se asemassa tekee?*
- *Mitä standardi suurinpiirtein on ja mitä se tarkoittaa työn kannalta?*
- *Kenelle työ tehtiin?*
- *Mikä on työn haluttu lopputulos ja tavoitteet?*
- *Mihin työ keskittyy kaikesta eniten?*
- *Mikä on työn tausta ennen työn aloittamista?*

Tämä diplomityö on tehty Alsus Oy:lle, työn tekohetkellä tekijän työpaikalle vuonna 2018. Tekijä valitsi työn aiheen mielenkiinnon ja ajankohdan sopiivuden takia. Työ liittyi sopivasti ajanhetkellä sen hetkisiin työtehtäviin.

Nykypäivänä sähköverkot ovat iso yhteiskuntaa ja sen sujuvaa toimivuutta. Ilman sähköä ei moni asia nykypäivänä toimisi niinkuin se on. Sähköä tarvitaan joka paikassa ja tietotekniikan lisääntyessä vieläkin enemmän. Nykypäivän sähköverkko koostuu useista erillisistä komponenteista, joita on mm. sähköntuontolaitos, sähkölinjat ja sähköasemat. Sähköasemien tehtävä verkossa on toteuttaa erilaisia toiminnallisuuksia, kuten muuntaja, jakaminen ja verkon toiminnan tarkkailu. Nykyisin sähköasemat eivät tarvitse henkilökuntaa paikalle, kuin huoltotehtävissä. Asemia toiminnallisuutta voidaan seurata etänä, vian tai huollon tarpeen sattuessa, huoltomies käy tarkistamassa asian paikanpäällä. Sähköaseman yksi tärkeä tehtävä on tarkkailla verkon toimivuutta ja vikatilanteen tapahtuessa esim. katkaista linjasta virrat pois vikatilanteen sattuessa. Vikatilanne voisi olla esim. kaapelin poikkimeno ja virta pääsisi tätä kautta maihin. Tässä tapauksessa sähköaseman katkaisee linjan virran ja linja jää odottamaan huoltoa.

Sähköasemien funktionaalisuutta nykypäivänä toteuttaa ns. älykkäät elektroniikka laitteet (engl. Intelligent Electronic Devices), lyhenteenä IED. IED:t voidaan kytkeä ja konfiguroida toteuttamaan monta eri funktionaalisuutta. Konfigurointi tapahtuu erillisillä konfigurointitiedostoilla. IED:t voivat kommunikoida ethernet-verkon yli aseman muun logiikan

ja muiden IED-laitteiden kanssa. Nykypäivänä verkon nopeus ja mahdollistaa reaaliaikaisen kommunikoinnin asemilla sen eri laitteiden välillä. IED voidaan esim. konfiguroida hoitamaan sähkölinjan kytkimenä oloa, joka myös tarkkailee linjan toimintaa mittaamalla konfiguroituja arvoja, kuten jännitettä ja virtaa. Vikatilanteen sattuessa IED katkaisee linjan enemmän vahingon välttämiseksi. Linjan korjauksen jälkeen virta kytketään takaisin päälle.

Monen eri toimijan toimiessa laitteita tuottavalla allalla ja sähköasema suuren elektronisen laitteiden määrän takia. On määritetty maailmanlaajuinen standardi IEC 61850 määrittämään koko aseman laitteiden välistä kommunikointiprotokollia varten. Standardi määrittää eri valmistajien IED-laitteille samat yhteiset kommunikointiprotokollat joita noudattamalla eri valmistajien laitteet sopivat yhteen.

Standardi määrittää mallit erilaisten viestien tilaukseen, jolla tilaaja voi tilata haluttuja konfiguroituja datapisteitä verkon yli. Tässä työssä keskitytään tämän asiakasohjelmiston suunnitteluun ja toteutukseen.

1.1 Tausta

Aikemmin mainitusta asiakasohjelmisto oli jo olemassa protoversio, joka pystyi tilaamaan, prosessoimaan ja tallentamaan saatuja viestejä. Ohjelmassa oli kuitenkin suorituskykyongelmia ja eikä se tukenut kaikkia standardin määrittämiä ominaisuuksia. Niinpä uudellelle asiakasohjelman toteutukselle oli tarve, joka myös korjaisi entisen toteutuksen ongelmat.

1.2 Laajuus

Työssä keskitytään uuden asiakasohjelman suunnitteluun toteutukseen tutkimusta teemmällä erilaisista arkkitehtuureista ja näistä valitsemalla parhaimman tilanteeseen sopien. Tutkimusta tehdään myös protoversion pohjalta selvittämään mikä aiheuttaa hitautta ja kuinka tämä voidaan ratkaista. Samalla käydään läpi entisen toteutuksen arkkitehtuuria, kuinka siitä saataisiin joustavampi ja se tarjoaisi myös laajennusmahdollisuuksia enemmänkin.

1.3 Tavoitteet

Tavoitteena työssä on suunnitella ja toteuttaa asiakasohjelmisto, joka kokonaan korvaa entisen protoversion ja toteuttaa kaikki standardin määrittämät toiminnallisuudet. Ohjelmiston pitäisi myös olla suorituskykyinen ja siinä ei saa olla samoja ongelmia kuin entisessä protoversiossa. Uuden toteutuksen pitäisi myös olla laajennettavissa uusille ominaisuuksille ja vaatimuksille.

1.4 Työn rakenne

Ensin työssä pohjustetaan toteutuksen ymmärtämiseen tarvittua teoriaa. Teorian jälkeen suunnitellaan itse toteutus ja argumentoidaan miksi tiettyihin valintoihin työssä päädyttiin. Suunnitelma pohjustaa tulevaa toteutusta ja sen arkkitehtuuria. Suunnittelun jälkeen tulee itse toteutus ja sen läpikäynti. Mitä kirjastoja ja tekniikoita toteutukseen on käytetty ja mitä mikäänkin tekee. Toteutuksen tarkoitus on tarjota lukijalle kuva toteutuksen tärkeistä rakenteista ja komponenteista. Lopussa tehdään tuloksien katselmointi ja tuloksia verrataan alussa asetettuihin tavoitteisiin. Tarkoituksena on antaa lukijalle kuva kuinka hyvin työn tavoitteisiin päästiin ja mitä olisi voinut tehdä toisin. Lopussa myös esitellään asioita jatkokehitystä ja tutkimusta varten.

2. TEORIA

Tähän kohtaan kirjoittaa teoriaa mitä tarvitaan työn toteutuksen ymmärtämisen kannalta. Kaikki työssä tarvittava teoria kuvataan tämän otsikon alla.

Tämän lukemalla lukija ymmärtää:

- Mikä on standardin ja sähköaseman relaatio ja kuinka ne liittyvät toisiinsa?
- Mikä on IEC standardi ja työn kannalta sen tärkeimmät piirteet.
- Kuinka viestien tilaus tapahtuu standardin mukaan?
- Kuinka standardi määrittää viestien rakenteen ja parameterit?
- Mikä on MMS-protokolla ja kuinka mallinnus standardin objekteista tehdään sille?

Tässä osiossa lukijaa perehdytetään työn kannalta tärkeään teoriaan. Teoriaosuuden kokonaan lukemalla lukija ymmärtää mitä IEC 61850 -standardi tarkoittaa sähköasemien kannalta ja mitä kaikkea se määrittää. Kuinka standardi määrittää tilattavat viestit ja mitä malleja ja palveluita niihin liittyy. Viestien malli ja toiminta on tärkeä ymmärtää työssä toteutetun ohjelman toiminnan kannalta. Työn lopullisessa toteutuksessa viestit prosessoitiin ja julkaistiin jonopalvelimelle myöhempää käyttöä varten. Teorian lopussa lukija perehdytetään jonopalvelimen toteutukseen liittyvään teoriaan. Myöskin tämän osuuden teoria on tärkeä ymmärtää ohjelman sisäisen toiminnan kannalta.

2.1 IEC 61850 -standardi

Kirjoita yleisesti mikä on IEC 61850 -standardi ja mitä varten se on olemassa. Kerro myös kuinka standardi on pilkottu pienempiin dokumentteihin ja mitä kukin käsittelee.

Sähköasemilla nykypäivänä käytössä olevilla älykkäillä elektronisilla laitteilla (engl. IED) toteutetaan aseman kontrollointi ja suojaus. Aseman suojauksen lisäksi siihen kuuluu myös asemalta lähtevät sähkölinjat. Jotta suojaus ja ohjaus olisi mahdollista täytyy aseman eri laitteiden kommunikoida verkon yli keskenään ja erilliselle kontrolliasemalle [1, s. 1]. Maailmanlaajuisesti määritetty IEC 61850 -standardi määrittää yhteiset säännöt eri valmistajien laitteiden välille kuinka kommunikointi toteutetaan [3, s. 10]. Ilman yhteisiä sääntöjä jokainen valmistaja olisi vapaa toteuttamaan omat säännöt ja protokollat. Seurauksena olisi, että laitteet eivät olisi keskenään yhteensopivia eri valmistajien välillä. Standardin on tarkoitus poistaa tämä ja määrittää yhteiset pelisäännöt kommunikoinnin toteuttamiseen aseman kommunikointiin liittyen eri laitteiden välillä [6, s. 1] .

Standardi on määritetty käytettäväksi myös tulevaisuudessa pohjana uusille toteutuksille. Standardi määrittää kaiken abstraktilla tasolla teknisestä toteutuksesta riippumatta. Tämän jälkeen abstraktit mallit ja palvelut voidaan mallintaa eri tekniikoille, joka mahdollistaa sen. [1, s. 2]

2.1.1 Standardin historia

Kirjoita tähän standardin historiaa ja kuinka se muodostui. Pohdi kuitenkin ensin onko tämä lukijan kannalta tärkeää ja tarvittavaa tietoa.

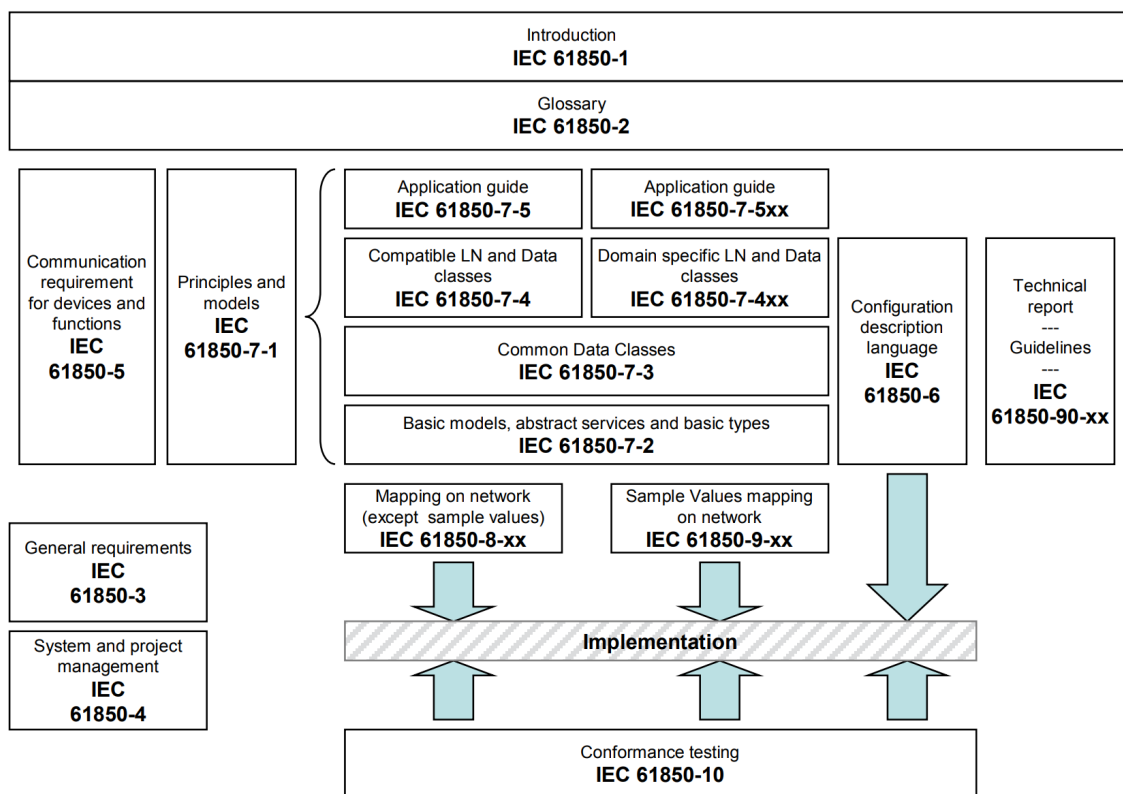
2.1.2 Standardin eri osat

IEC 61850 -standardi on todella laaja kokonaisuus. Tämän takia se on pilkottu erillisiin dokumentteihin, josta jokainen käsittelee tiettyä asiaa. Historian saatossa standardiin on lisätty uusia dokumentteja laajentamaan standardia [5, 8]. Tämän työn kirjoitushetkellä standardiin kuului lisäksi paljon muitakin dokumentteja, esimerkiksi uusiin mallinnuksiin muille tekniikoille ja vesivoimalaitoksien mallintamiseen liittyviä dokumentteja. Laajuudesta huolimatta standardin voi esittää 10:llä eri pääkohdalla ja näiden alakohdilla. Taulukossa 1 on esitetty standardin pääkohdan dokumentit ja niiden alkuperäiset englanninkieliset otsikot [7, s. 2] [5]. Kuvassa 1 on myös esitetty standardin eri osat ja niiden väliset relaatiot.

Taulukko 1. IEC 61850 -standardin pääkohtien ja niiden alakohtien dokumentit.

Osa	Otsikko englanniksi
1	Introduction and overview
2	Glossary
3	General requirements
4	System and project management
5	Communication requirements for functions and device models
6	Configuration description language for communication in power utility automation systems related to IEDs
7-1	Basic communication structure - Principles and models
7-2	Basic information and communication structure - Abstract communication service interface (ACSI)
7-3	Basic communication structure - Common data classes
7-4	Basic communication structure - Compatible logical node classes and data object classes
8-1	Specific communication service mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3
9-2	Specific communication service mapping (SCSM) - Sampled values over ISO/IEC 8802-3
9-3	Precision time protocol profile for power utility automation
10	Conformance testing

Standardin ensimmäiset osat 1–5 kattavat yleistä kuvaa standardista ja sen vaatimuksista. Osiossa 6 käsitellään IED-laitteiden konfigurointiin käytetty XML (engl. Extensible Markup Language) -pohjainen kieli [2, s. 7–8]. Tämä osuus ei ole tämän työn kannalta tärkeä



Kuva 1. IEC 61850 -standardin osat ja niiden väliset relaatiot [3, s. 14].

ja sitä ei sen tarkemmin käsitellä. Osat 7.1–7.4 käsittelevät standardin abstraktia mallia ja kuinka se rakentuu. Tämä lyhennetään standardissa ACSI (engl. Abstract Communication Service Interface), ja samaa lyhennettä käytetään tässä työssä. Osissa 8–9 ja niiden alakohdissa käsitellään abstraktimallien mallintamista erillisille protokollille, jolloin malleista tulee kyseisestä tekniikasta riippuvaisia. Abstrakteja malleja ja niiden mallintamista tekniikalle käsitellään teoriassa tuonnempana. Osa 10 käsittelee testausmenetelmiä, joilla voidaan varmistaa standardin määritysten noudattaminen. Tämä osuus ei myöskään ole tämän työn kannalta tärkeä ja sitä ei teoriassa sen takia käsitellä.

2.1.3 Standardin määrittämä abstraktimalli

Kirjoita tähän mitä standardin IEC 61850-7-2 osuudessa määritellään abstraktoimalla fyysisiä laitteita ja palveluita rajapinnoiksi ja olioiksi. Käsittelee standardin Abstract communication service interface (ACSI).

IEC 61850 -standardin määrittämät konseptit ja abstraktimallit voidaan käyttää myös mallintamaan:

- vesivoimalaitos,
- sähköasemien välistä kommunikointia,
- informaation välitys hajautetun automaation välillä,

- informaation välitys sähköaseman ja ohjauskeskuksen välillä,
- informaation välitys mittaukseen,
- tilan monitorointia ja diagnosointia, ja
- informaation välitys konfigurointiin tarkoitettulle tekniikalle [3, s. 11].

2.1.4 Viestiblokin konfigurointi ja tilaus

Kirjoita tähän IEC 61850 -standardin määrittämästä abstraktista raportointimallista. Tätä raportointi mekanismia tullaan käyttämään raporttien tilauksessa ja sen konfigurointi täytyy ymmärtää toteutettavan ohjelmiston kannalta.

2.1.5 Viestin rakenne

Kirjoita tähän standardin määrittämästä viestin rakenteesta ja mitä tietoa se sisältää. Kerro myös sen vaihtoehtoisista kentistä.

2.2 Abstraktimallin sovitus MMS-protokollaan

Kirjoita kuinka ylempi ACSI sovitetaan MMS-protokollan palveluiksi ja tietotyypeiksi standardin IEC 61850-8-1 osuuden mukaan. Tähän myös miten raportointi toimii MMS-protokollan päällä.

2.2.1 MMS-protokolla

Selitä lyhyesti mikä on MMS-protokolla ja vähän sen tietotyypeistä. Tämän tarkoitus on pohjustaa tulevaa IEC 61850 abstraktien olioiden (ACSI) sovitusta tämän protokollan päälle.

2.3 Advanced Message Queuing Protocol

Kirjoita tähän AMQP määrittävästä standardista, mikä sen tarkoitus on ja mihin sitä voidaan käyttää.

2.3.1 Viestien välitysmekanismit

Mitä mekanismeja AMQP tarjoaa viestien välittämiseen osapuolille. Näitä on jono, reititys suoraan osapuolien välillä ja viestin julkaisu ja tilaaminen.

2.3.2 Tilaus ja julkaisu -mallin osat

Kirjoita tähän AMQP tarjoamista viestien julkaisu ja tilaus -mallin osista osapuolten kesken. Kerro mitä eri osat tekevät ja mikä niiden tehtävä viestien välittämisessä on. Englanniksi osia ovat esim. exchange, queue, publisher ja consumer.

3. ALKUTILANNE

Pohjista miksi suunniteltava ohjelmisto tarvitaan toteuttaa yritykseen johon työn teen. Alustava suunniteltu ohjelmiston toteutus olisi tilata IEC 61850 -standardin määrittämiä raportteja ja muokata ne uuteen muotoon ja julkaista ne eteenpäin tilaavalle ohjelmalle käyttäen AMQP-standardin määrittämää viestintää. Jonon tilaava asiakasohjelmisto voi olla mikä tahansa muu ohjelmisto. Viestien lopullinen muoto voisi olla JSON.

Nykyisin sähköasemilla älykkäillä elektronisilla laitteilla (engl. *Intelligent Electronic Devices*, **IED**) asemilla voidaan toteuttaa tuhansia eri datapisteitä, jotka kuvaavat aseman toiminnallisuutta ja konfiguroitavuutta. Tämän konfiguroitavuuden ansiosta IED:tä voidaan asemalla käyttää erilaisina sähkölaitteina, kuten sulakkeina tai muuntajina. IEC 61850 -standardin abstraktit datamallit määrittävät IED:n datapisteiden rakenteet, muodot ja tyytit. Standardin mukaan erillisistä datapisteistä voidaan muodostaa haluttuja datajoukkoja (engl. *data set*). Datajoukkot ovat helppo tapa kuvata halutut tai tärkeät datapisteet yhdeksi yhteinäiseksi joukoksi. [7].

Asiakas-palvelin-arkkitehtuurissa asiakkaan on mahdollista tilata datajoukkojen sen hetkisiä arvoja IEC 61850 -standardin määrittäminä raportteina konfiguroitavilla parametreilla, jotka konfiguroidaan ennen tilausta. Arkkitehtuurissa asiakas tekee tilauksen palvelimelle (tässä tapauksessa IED), jonka jälkeen palvelin lähettää raportteja asiakkaalle automaattisesti, jonkin asiakkaan konfiguroiman ehdon täytyessä. Standardi määrittää kuinka raportteja voidaan esim. välittää TCP/IP-protokollan avulla. Yksi raportti sisältää mm. tietojoukon sen hetkisiä arvoja ja syyn raportin lähetykseen (esim. arvon muuttuminen). Palvelin ylläpitää tilausta kunnes asiakas lopettaa tilauksen tai yhteys osapuolten välillä katkeaa. Asiakas tilaa raportit konfiguroimalla palvelimella olevan erillisen raportointilohkon (engl. *Report Control Block*, **RCB**). Lohkolla voi konfiguroida mm. raporttien sisältämiä vaihtoehtoisia kenttiä ja erilaisia liipaisimia raporttien generointiin. Standardi määrittää että yhdellä RCB:llä voi olla vai yksi tietojoukko ja yksi tietojoukko voi olla viitattuna monessa eri RCB:ssä. Yhdessä IED:ssä voi olla määritetynä monta RCB:tä. Yhtä tilaavaa asiakasta kohden on yksi RCB instanssi. [4, s. 91–130].

Tulevissa kappaleissa pohjustetaan työn alussa olemassa olevan ohjelmiston arkkitehtuuria, mitkä olivat sen komponentit ja niiden toiminnallisuus. Tämän jälkeen pohditaan toteutuksen ongelmia, ja mitä työssä pyritään ratkaisemaan uudella toteutuksella. Asetettujen tutkimuskysymysten ja ongelmien kautta pyritään löytämään uudelle ohjelmiston arkkitehtuurille pohjaa ja ratkaisua siihen liittyviin päätöksiin.

3.1 Kokonaiskuva

Kirjoita tähän osioon kokonaiskuva alkutilanteesta missä oltiin ennen työn aloittamista. Selvennä kuvilla alkutilanteen arkkitehtuuria.

Työn aloitusvaiheessa oli jo toteutettuna ohjelmisto raporttien tilaukseen ja käsittelyyn. Tämä toteutus oli puutteellinen, ei helposti skaalautuva, ja huono suorituskäytännön. Alkuperäinen ohjelmisto oli lähellä enemmän ensimmäistä prototyyppiä ennen todellista toteutusta. Työn tarkoituksena oli suunnitella ja toteuttaa uusi toteutus, joka ratkaisisi entisen ongelmakohdat.

Alkuperäisessä toteutuksessa asiakasohjelmisto oli toteutettu Ruby-ohjelmointikielellä. IEC 61850 -standardin määrittämien palveluiden ja tietorakenteiden toteutukseen käytettiin avoimen lähdekoodin libIEC61850-kirjastoa¹. Kirjasto on ohjelmoitu C-kielellä ja sen avulla voidaan toteuttaa kumpikin palvelin- ja asiakasohjelmisto. Tässä toteutuksessa tarvittiin vain asiakasohjelman osuutta Ruby-osuuden toteutukseen. Kirjasto abstraktoi standardin määrittämiä palveluita ja tietorakenteita ohjelmoijalle helpoiksi funktioiksi ja C-kielen rakenteiksi. Normaalisti C-koodin funktioiden kutsuminen Rubysta suoraan ei ole mahdollista ilman erillistä liitosta. Seurauksena Rubyyn oli tehty laajennos libIEC61850-kirjastoon käyttäen Rubylle saatavaa ruby-ffi -kirjastoa² (engl. *Foreign Function Interface*, **FFI**). Liitoksen avulla libIEC61850-kirjasto voi hoitaa standardin vaatiman matalan tason toiminnan ja Ruby-ohjelmisto voi keskittyä vaadittuun toiminnallisuuteen.

Kirjasto toteuttaa raporttien vastaanoton palvelimelta erillisellä säikeellä. Säie käynnistetään kun asiakasohjelma asettaa funktion takaisinkutsuntaa varten raportin saapuessa ja aloittaa tilauksen. Asetettua funktiota kutsutaan asynkronisesti erillisestä säikeestä raportin saapuessa asiakkaalle. Takaisinkutsun suorituksen jälkeen, suoritus palaa takaisin säikeeseen.

3.2 Ratkaistavat ongelmat

Kirjoita tähän mitä ongelmia edellisen toteutuksen kanssa on ja mitä yritään ratkoa. Mainitse suorituskäytännön ongelmista Rubylla ja libiec61850-kirjastoa käyttäen.

Työn alussa olevan ohjelmiston ongelmia oli mm. ei helposti skaalautuvuus, huono suorituskäytännön raporttien määrän ollessa suuri, eikä ohjelmisto tukenut kaikkia standardin määrittämiä toiminnallisuuksia. Ohjelmistoa voisi enemmän pitää ensimmäisen toteutuksen prototyyppinä. Ohjelman suoritusalueena käytettiin Linuxia.

Ohjelmisto pystyi tilaamaan ja vastaanottamaan raportteja yhdeltä IED:ltä ja siinä monelta määritellyltä RCB:ltä. Ohjelmisto prosessoi ja tallensi raportteja tietokantaan muuta

¹<http://libiec61850.com>

²<https://github.com/ffi/ffi>

käyttöä varten. Tilanteessa, jossa raportteja tilaavassa järjestelmässä on monta osaa, jotka kaikki tarvitsevat raporttien tietoja reaaliajassa. Joutuvat eri osat tässä tilanteessa kyselemään tietoja tietokannasta, ilman erillistä tietoa niiden saapumisesta. Tämä aiheuttaa turhaa kuormaa tietokannalle ja tietojen saaminen reaaliajassa ei ole mahdollista. Myöskin jos komponentti tarvitsee tietyn tyyppin raportteja, ei kaikkea tietoa, ongelma on sama.

Ohjelmiston suorituskyky paikoin raporttien määrän ollessa suuri aiheutti ongelmia. Syynä Rubyn toteutuksessa oli oletustulkissa (*CRuby*) oleva globaali lukitus (engl. *Global Interpreter Lock*, **GIL**). Vaikka Rubyn säie on oma käyttöjärjestelmän tarjoama säie, GIL estää säikeiden yhtäaikaisen suorituksen ja vain yksi säie on suorituksessa kerrallaan [9, s. 131–133]. Linux-pohjaisella käyttöjärjestelmällä libIEC61850-kirjaston laitteistoabstraktiokerros (engl. *Hardware Abstraction Layer*, **HAL**) käyttää POSIX-säikeitä [10]. Linux-käyttöjärjestelmän säikeet ovat suorituksessa yhtä aikaa ja moniytimisellä prosessoreilla asioita tapahtuu samalla ajan hetkellä. Nyt raportin saapuessa, C-prosessin säikeen suoritus kutsuu takaisinkutsuntaan asetettua funktiota, joka on implementoitu Rubyn puolella. On funktion suoritus GILin alaista suoritusta. Ruby-prosessin myös suoritustaessa muuta toimintaa takaisinkutsujen välissä, on Rubyn suorituskyky ohjelmiston pulonkaulana raporttien määrän ollessa tiheää.

Kirjoita tähän vielä ongelmasta kun tilataan monta RCB:tä. Raporttien tullessa Rubyn puolelle, ei Rubyn muu koodi saa tilattua loppuja RCB:tä kirjaston lukitusten takia. Ja yhteys aikakatkeaa tämän takia. Selitä lukituksista tarkemmin ja myös liitä pätkiä libIEC61850-kirjaston koodista. Syyn selityksen voi siirtää muualle. Kirjoittaa vain että on ongelma, ja selvitys miksi, muualla.

3.3 Tutkimuskysymykset

Esitä tässä työlle asetettuja tutkimuskysymyksiä. Näitä voisi olla esim. seuraavat:

- *Mikä on syynä huonoon suorituskykyyn alkutilanteen toteutuksella?*
- *Kuinka suorituskyky paremmaksi verrattuna nykyiseen toteutukseen?*
- *Mitkä ohjelmiston arkkitehtuurin suunnittelumallit (design patterns) olisivat sopivia tämän kaltaisen ongelman ratkaisemiseen? Mitä niistä pitäisi käyttää ja mitä ei?*
- *Mikä olisi sopiva lopullisen prosessoidun tiedon muoto?*
- *Kuinka järjestelmä hajautetaan niin että tiedon siirto eri osapuolten välillä on mahdollista ja joustavaa (push vs pull, message queue jne.)?*

4. SUUNNITTELU

Kirjoittaa tähän kuinka toteutettava arkkitehtuuri suunniteltiin ja kuinka päätöksiin päädyttiin. Kirjoitusta myös miten tilattuja raportteja käsitellään ja kuinka niitä julkaistaan eteenpäin. Tarkoituksena olisi saada raportit nykyaikaiseen JSON muotoon.

4.1 Suorituskyvyn parantaminen

Miksi entisen toteutuksen suorituskyky on huono ja mitä voitaisiin tehdä sen parantamiseksi. Kirjoita vaikutuksista tähän ja mihin tuloksiin päädyttiin.

4.2 Järjestelmän hajautus

Lähde erilaisista hajautuksista (pull vs push, message queue) ja päätä mikä sopii tähän toteutukseen parhaiten ja miksi.

4.3 Ohjelmiston parametrisointi

Kirjoita mitä asiakasohjelman pitää tehdä jotta raportit saadaan tilattua ja mitä parametrejä ohjelmisto tarvitsee toimiakseen. Kuinka käyttäjä kontrolloi ohjelman eri asetuksia?

4.4 Arkkitehtuurin suunnittelu

Määritä ohjelman tarkempaa arkkitehtuuria mitä voidaan käyttää asetettujen ja yllämainittujen asioiden saavuttamiseen ja tarkentamiseen. Mitä jos käyttäjä tilaa monta viestiblokkia, niin missä järjestyksessä asiat tehdään jne.

4.5 Prosessoidun viestin muoto

Kirjoita tähän mihin muotoon viestit lopussa tallennetaan esim. JSON. Miksi tähän valintaan päädyttiin. Kerro myös kuinka raportin alkuperäistä rakennetta muokattiin uuteen muotoon sopivaksi.

5. TOTEUTUS

Kirjoita tähän osioon siitä kuinka suunniteltu arkkitehtuuri toteutettiin ja millä tekniikoilla. Tämä osio käyttää lyhyitä koodiesimerkkejä hyväkseen selittämään lukijalle kuinka toteutus tehtiin, jotta lukija voisi itse toteuttaa samanlaisen ohjelmiston.

5.1 Ohjelmiston toteutuksen valinta

Kirjoita tähän miksi päädyttiin tietynlaiseen ohjelmiston toteuttamiseen. Työssä on mietitty komentorivipohjaista toteutusta. Lisäksi mille alustalle ohjelmisto suunnitellaan Windows vai Linux.

5.2 Kielen valinta

Kirjoita tähän mikä kieli valittiin toteutuksen tekemiseen ja miksi tämä. Alustava suunnitelma on toteuttaa C-kielillä.

5.3 RabbitMQ

Kirjoita tähän RabbitMQ toteutuksesta. Kirjasto toteuttaa AMQP-standardin määrittämiä eri viestintämalleja. Kerro kuinka sitä hyödynnetään tässä työssä ja vähän sen että mitä vaatii.

5.4 Käytettävät kirjastot

Kirjoita tähän erilaisista kirjastoista mitä toteutukseen valittiin ja miksi. Alaotsikoita voi lisätä jos toteutukseen tarvitaan muita kirjastoja.

5.4.1 libiec61850

IEC 61850 -standardin toteuttava C-kirjasto joka tekee raskaan työn standardin määrittämien palveluiden toteuttamiseen ja muodostamiseen. Kirjasto tarjoaa rajapinnat serveri- ja asiakasohjelmiston toteuttamiseen, mutta vain asiakasohjelmiston rajapintoja käytetään. Kirjasto tarjoaa myös rajapinnat haluttujen raporttien tilaamista varten. Kirjaston nettisivu täältä: <http://libiec61850.com/libiec61850/>.

5.4.2 rabbitmq-c

RabbitMQ:n rajapinnan toteuttava kirjasto C-kielen ohjelmille. Kirjastolla voidaan toteuttaa julkaisevia ja tilaavia ohjelmistoja. Kirjastosta käytetään julkaisevan puolen toteutusta. Kirjasto löytyy täältä: <https://github.com/alanxz/rabbitmq-c>.

5.4.3 JSON-formatointi

Joku kirjasto JSON formatointiin C-kielelle. Näkyy olevan parikin vaihtoehtoa. Perustele tähän valinta ja miksi.

5.5 Jatkokehitys

Kirjoita tähän ideoita mitä jää jatkokehitykseen ja mitä ohjelmistossa on puutteita tai mitä jäi tekemättä.

6. ARVIOINTI

Kirjoitta tähän arviota työn tuloksista.

7. TULOKSET

Kirjoita tähän lopputuloksen analysoinnista ja peilaa saatuja tuloksia työlle alussa asetettuihin kysymyksiin. Mitä jäi saavuttamatta, mitä saavutettiin ja miten hyvin? Mitä olisi voinut parantaa? Voi jakaa aliotsikoihin jos tarvetta.

8. YHTEENVETO

Kirjoita tähän ensin arviointi ja yhteenveto työstä ja sen lopputuloksista. Mitä hyötyjä työnantaja työstä saa ja jatkokehitysideoita. Mitä työssä meni hyvin ja mitä olisi voinut tehdä toisin?

Kommentteja työtä aloittaessa:

- *Olisiko hyvä, että lähdet työssäsi erilaisista hajautus paradigmoista (push vs pull; message queue), perustelet valintasi ja sitten menet suunnitteluun ja toteutukseen?*
- *Ja olisi hyvä, että työ perustelee miksi tuota MQ arkkitehtuuria yleensä (ja rabbitMQ:ta) käytetään.*

Things to do now:

- *Laittaa aihe hyväksyntään.*
- *Lähde kirjoittamaan teoriaa ja ennen sitä yleistä tasoa missä ollaan. Yleinen korkea taso sen takia, että lukija ymmärtää mistä edes on kyse. Pidä koko ajan kirjoittaessa mielessä top-down lähestymistapa! Erittäin tärkeä!!!*
- *Loppu otsikoida niin että ensin on tulokset, niiden arviointi ja yhteenvedo mainituissa järjestyksessä.*
- *Kirjoittaessa miettiä asioita mistä kirjoitetaan ja pitää kontekstista kiinni.*
- *Pidä lauseet simppelineinä ja helppolukuisina! Älä turhaan vaikeuta hommaa lukijalle ja se ei tuo työhön yhtään mitään lisäarvoa! Todella tärkeä asia ajatella! Jos lause käsittää monta asiaa, pilko se pienempiin erillisiin lauseisiin.*
- *Muihinkin lähteisiin voi viitata kuin tieteellisiin. Toki yritä löytää tieteellisiä julkaisuja mahdollisuuksien mukaan. Osoittaa että olet perehtynyt asiaan paremmin.*
- *Kun kirjoitat asiaa esim. että entisessä ohjelmassa oli ongelma että ei skaalaudu helposti tai on huono suorituskyky. Kerro mistä johtopäätös tulee. Tämä ei ole lukijalle selvää tietoa.*
- *Teorien ja yleisen osuuden kirjoittamisen jälkeen, sovi palaveri Karin kanssa.*
- *Työn otsikko on hyvä, ei tarvitse olla erikseen "ohjelmallisesti"sanaa.*
- *Työn päätason otsikoita laittaa enemmän kuvaavimmiksi kuin "Alkutilanne"ja "Teoria".*
- *Käytä työssä viesti sanaa raportin sijaan. Tuo lukijalle esille että se tarkoittaa standardin mukaisia raportteja.*

Huomioituja asioita toisten dipoissa:

- *Tärkeät sanat esitellään tekstissä ensimmäisen kerran kursiivilla painottamisen takia. Tämän jälkeen ei enää samaa sanaa kursivoida.*
- *Todella paljon erilaisia lähteitä käytetty! Blogiposteja, kirjoja, ja tapahtumien kirjoituksia (IEEE). On myös nettisivuja käytetty lähteenä kun mainitaan esim. Git ja jotain muita sivuja. Nämä tietysti voi olla myös alaliitteenä sivulla.*
- *Tosi hyvin kirjoitettu! Todella selkeää tekstiä ja etenee hyvin ja on lukijalle ystävällinen.*

- *Johdanto on pilkottu otsikoihin työn alkutilanteen selvittämiseksi hyvin ennen teoriaa. Ja teoriaosuus alkaa joustavasti johdannon jälkeen järkevästi.*
- *Kun listataan tekstiä, sana on ensin kurstiivilla ja on selitetty asiaa. Kohta loppuu puolipisteeseen (;). Tämän jälkeen jatkuu pienellä seuraava aihe ja päättyy myös puolipisteeseen. Viimeinen kohta alkaa myös pienellä, mutta päättyy pisteeseen normaalisti. Seuraava kappale alkaa normaalisti. Listassa lauseet muokkautuvat yhteen esim. käyttäen ja sanaa.*
- *Kysymys teoriassa mikä työssä on jäljellä oli kirjoitettu kurstiivilla.*

LÄHTEET

- [1] C. Brunner, IEC 61850 for power system communication, teoksessa: 2008 IEEE/PES Transmission and Distribution Conference and Exposition, April, 2008, s. 1–6.
- [2] IEC 61850-6 Communication networks and systems for power utility automation – Part 6: Configuration description language for communication in electrical substations related to IEDs, International Electrotechnical Commission, International Standard, Dec. 2009, 215 p. Saatavissa: <https://webstore.iec.ch/publication/6013>
- [3] IEC 61850-7-1 Communication networks and systems in substations - Part 7-1: Basic communication structure for substation and feeder equipment - Principles and models, International Electrotechnical Commission, International Standard, July 2003, 110 p. Saatavissa (viitattu 16.5.2018): <https://webstore.iec.ch/publication/20077>
- [4] IEC 61850-7-2 Communication networks and systems for power utility automation - Part 7-2: Basic information and communication structure - Abstract communication service interface (ACSI), International Electrotechnical Commission, International Standard, Aug. 2010, 213 p. Saatavissa (viitattu 16.5.2018): <https://webstore.iec.ch/publication/6015>
- [5] IEC 61850:2018 SER Series, International Electrotechnical Commission, verkkosivu. Saatavissa (viitattu 9.6.2018): <https://webstore.iec.ch/publication/6028>
- [6] K. Kaneda, S. Tamura, N. Fujiyama, Y. Arata, H. Ito, IEC61850 based Substation Automation System, teoksessa: 2008 Joint International Conference on Power System Technology and IEEE Power India Conference, Oct, 2008, s. 1–8.
- [7] R. E. Mackiewicz, Overview of IEC 61850 and Benefits, teoksessa: 2006 IEEE PES Power Systems Conference and Exposition, Oct, 2006, s. 623–630.
- [8] New documents by IEC TC 57. Saatavissa (viitattu 9.6.2018): <http://digitalsubstation.com/en/2016/12/24/new-documents-by-iec-tc-57/>
- [9] R. Odaira, J.G. Castanos, H. Tomari, Eliminating Global Interpreter Locks in Ruby Through Hardware Transactional Memory, SIGPLAN Not., Vol. 49, Iss. 8, Feb. 2014, pp. 131–142. Saatavissa (viitattu 16.5.2018): <http://doi.acm.org/10.1145/2692916.2555247>

- [10] Official repository for libIEC61850, the open-source library for the IEC 61850 protocols <http://libiec61850.com/libiec61850>, GitHub verkkosivu. Saatavissa (viitattu 17.5.2018): <https://github.com/mz-automation/libiec61850>

- [11] B.D. Stockton, Design Guide for Rural Substations, United States Department of Agriculture, June 2001, 763 p. Saatavissa (viitattu 25.5.2018): https://www.rd.usda.gov/files/UEP_Bulletin_1724E-300.pdf