

Aktiivisten antennien kalibrointi- ja testausjärjestelmän ohjelmistoarkkitehtuuri

Leevi Sulonen

OPINNÄYTETYÖ
Marraskuu 2020

Tieto- ja viestintäteknikka
Sulautetut järjestelmät

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintäteknikka
Sulautetut järjestelmät

SULONEN, LEEVI

Aktiivisten antennien kalibrointi- ja testausjärjestelmän ohjelmistoarkkitehtuuri

Opinnäytetyö 24 sivua, joista liitteitä 2 sivua
Marraskuu 2020

Tämä opinnäytetyö tehtiin DA-Group -konsernin AACT tutkimushankkeeseen (Active Antenna Calibration and Test). Opinnäytetön aiheena oli ohjelmistoarkkitehtuurin luominen aktiivisten antennien kalibrointi- ja testausjärjestelmään. Tutkimushankkeen omistaa ja rahoittaa Euroopan avaruusjärjestö ESA.

Lähivuosina on kasvanut tarve laajakaistaisille sähköisesti ohjattaville liikkuville antennimoduleille, joita voidaan käyttää maalla, meressä ja ilmassa. Jotta tällaisten antennien suorituskyky voidaan varmentaa, ne pitää mitata, kalibroida ja testata tuotannon yhteydessä. Tällaisia järjestelmiä on olemassa, mutta ne ovat hitaita ja epäkäytännöllisiä. Projektin tarkoitus oli luoda järjestelmä, joka kykenee mittaamaan antennit pienehkössä ajassa kaikuisassa ja häiriöisessä tuotantoympäristössä. Opinnäytetyö tehtiin luomalla UML-kaltaisia kaavioita. Opinnäytetyön lopputulos on korkealla abstraktiotasolla kuvattu toimiva järjestelmä.

Asiasanat: ohjelmisto, antenni, vastaanotinmatriisi, kalibrointi

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Information and communication technology
Embedded systems

LEEVI SULONEN

Software architecture for Active Antenna Calibration and Test system

Bachelor's thesis 24 pages, appendices 2 pages
November 2020

This thesis was done in the AACT research project for DA-Group. The subject of the thesis was to create a software architecture for Active Antenna Calibration and Test system. The research project is owned and funded by European Space Agency ESA.

A need for broadband electrically steered antennae operating at land sea and air has grown in the recent years. Such antennae require testing and calibration in order to verify the operational capabilities. Several systems capable of verifying the performance exist, but they are slow and impractical. The purpose of the research project was to create a system that is able to measure active antennae rapidly in echoic and electromagnetically noisy production environment with reasonable physical dimensions. The thesis was done by creating UML-like drawings. The end-result of the thesis is a functional system described in high level abstraction.

Key words: software, antenna, receiver matrix, calibration

SISÄLLYS

1	JOHDANTO	6
2	OHJELMISTOARKKITEHTUURI	7
	2.1 Ohjelmistoarkkitehtuurin standardi ISO/IEC 42010	8
3	AACT -järjestelmä.....	10
	3.1 Järjestelmän kalibrointi.....	10
	3.2 Antennin mittaus ja kalibrointi	12
	3.3 Data- ja hallintaluokat.....	14
	3.4 Rajapinnat.....	14
	3.4.1 AUT	15
	3.4.2 FF-TP	16
	3.4.3 NF-FF	17
	3.4.4 NF-skanneri.....	18
	3.4.5 RFLO.....	19
	3.4.6 VSA	20
4	POHDINTA	21
	LÄHTEET	22
	LIITTEET	23
	Liite 1. Järjestelmäkalibrointi - täysikokoinen	23
	Liite 2. Antennin mittausproseduuri - täysikokoinen.....	24

ERITYISSANASTO

AACT	Active Antenna Calibration and Test
TX	Lähetin
RX	Vastaanotin
PWS	Tasoaallon spektri (Plane Wave Spectrum)
TP	Takaprojektio (Back Projection)
VSA	Vektorisignaalianalysointilaite
RFLO	Radiotaajuinen paikallisoskillaattori (RF local oscillator)
SNR	Signaali-kohina -suhde (signal to noise ratio)
UML	Unified Modeling Language, mallinnuskieli
AUT	Antenna Under Test

1 JOHDANTO

Nykymaailmassa mahdollisuus liikkua on kasvanut siitä, mitä se on ollut. Maalla, merellä ja ilmassa on enenevässä määrin ihmisiä, jotka tarvitsevat internetyhteyden. Aikojen saatossa kyky paikantaa itsensä (GPS, GLONASS) on parantunut, mutta liikkuva internetyhteys on vielä hyvin paljon riippuvainen paikallisten palveluntarjoajien verkon peittävydestä. Jos yksilö on kaukana tukipisteistä, yhteyttä verkkoon ei välttämättä ole ollenkaan. Tätä varten on kehitteillä satelliittikommunikaatioon perustuva tietoinfrastruktuuri, jolla voidaan mahdollistaa laajakaistainen internetyhteys etäisimpiin maailman kolkkiin.

Satelliittiyhteydet toimivat usein hyvin korkeilla taajuuksilla ja tukipisteet ovat liikkuvia, joka tarkoittaa sitä, että tavanomaiset lähetin-vastaanotinjärjestelmät (staattiset antennit, staattiset tukipisteet etc.) eivät kykene kyseiseen kommunikaatioon. Tämän takia useat antennivalmistajat ovat kehittäneet sekä kehittävät jatkuvasti vaiheistettuja mikroaaltoantenneja. Nämä antennit pystyvät ohjaamaan säteily- ja vastaanotinkiilaansa sähköisesti, mahdollistaen nopeat muutokset säteilykulmassa (elevaatio, atsimuutti) joilla satelliitteihin saadaan yhteys. Kuitenkin, näiden antennien toimintaa on vaikea varmentaa tuotantoympäristössä, sillä nykyaikaiset mittausjärjestelmät ovat hitaita ja vaativat paljon tilaa. Ratkaisuksi tähän ongelmaan DA-Group kehitti järjestelmän, joka pystyy mittaamaan vaiheistettujen antennien lähikentän nopeasti, analysoimaan mittausdataa, sekä laskemaan korjauskertoimet mitattavien antennien apertuureille, jotta lähetys- ja vastaanotinkuviot saadaan eheiksi.

2 OHJELMISTOARKKITEHTUURI

Tässä kappaleessa käsitellään ohjelmistoarkkitehtuureja: mitä ne ovat, sekä mikä merkitys niillä on järjestelmäsuunnittelun kannalta. Nimi *ohjelmistoarkkitehtuuri* on periytynyt rakennuksien ja maisemien ilmeiden ja rajapintojen suunnittelusta, arkkitehtuurista.

2.1 Ohjelmistosuunnittelu vs. ohjelmistoarkkitehtuurisuunnittelu

Jo muinaisina aikoina Charles Babbage loi laskentakoneen, joka oli ohjelmoitavissa. Ada Lovelace kirjoitti siihen ensimmäisen ohjelman, ja häntä pidetään maailman ensimmäisenä ohjelmoijana. Ohjelman sisältö oli yksinkertainen, mutta sen merkitys oli hyvin suuri: koneisiin voidaan hengittää ”elämää” antamalla niille käskyjä.

Jo maailmansotien aikaan oli olemassa ohjelmoitavia koneita, mutta ne olivat hyvin suuria ja kalliita. Vasta transistorin keksimisen jälkeen ja integroitujen piirien yleistyessä tietokoneet ja niiden ohjelmointi lisääntyivät, sillä hinnat putosivat fyysisten kokojen mukana. Vanhoina aikoina ohjelmat olivat pieniä, ja ne kirjoitettiin matalan tason kielillä. Näiden ohjelmien lähdekoodia oli helppo analysoida, varsinkin, jos se oli kommentoitu hyvin. Nykypäivänä ohjelmistot ovat muuttuneet niin monimuotoisiksi ja laajoiksi (poislukien *ad hoc* ohjelmat etc.), että niitä on vaikea suunnitella ilman ”pohjapiirustuksia”, eli arkkitehtuuria. Hyvä arkkitehtuurisuunnittelu lisää ohjelmistosuunnittelun helppoutta, järjestelmien yhteenliittoa, yhteensopivuutta ja kehitettävyyttä (ISO 42010, v).

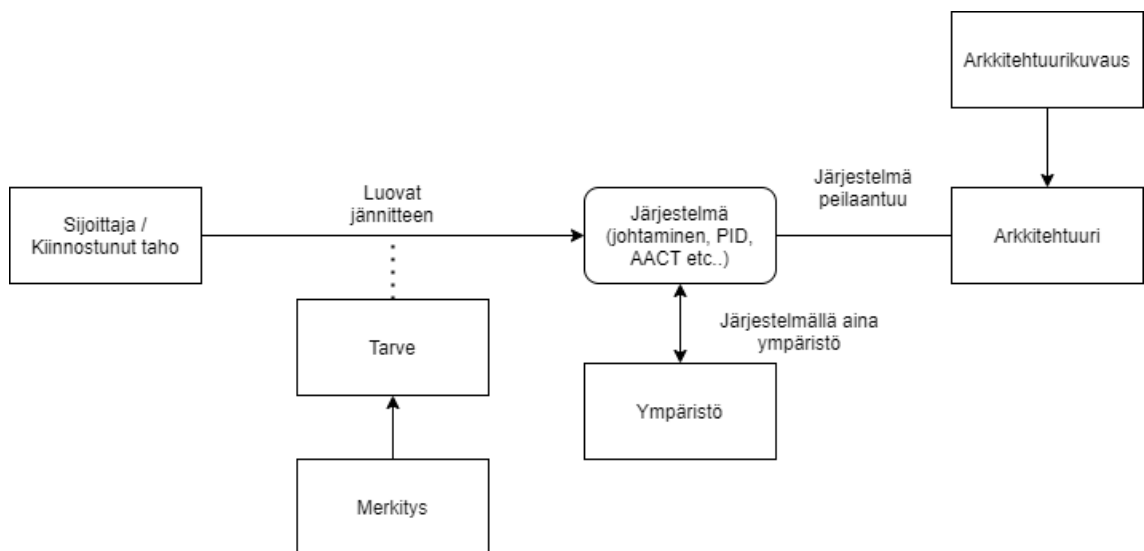
Arkkitehtuurisuunnittelu kuvaa tietyn ohjelman/ohjelmistokokonaisuuden jäsenten (laitteet, ohjelman moduulit) funktioita, niiden välistä kommunikaatiota, sekä kommunikaation protokollia. Ohjelmistosuunnittelu tarkoittaa ohjelmakoodin palasten toteuttamista (esimerkiksi käytetäänkö jossain kohdassa *for*- vai *while* silmukkaa) yksityiskohtaisella tasolla. Ohjelmistoarkkitehtuurisuunnittelussa on ohjelmistoseikkojen lisäksi elementtejä järjestelmäsuunnittelusta. Periaatteiltaan ne ovat hyvin samanlaisia. Analogiaa voidaan hakea rakennustekniikasta, jossa

arkkitehti luo yleisilmeen ja rajapinnat rakennuksille, ja jonka yksityiskohtaisen toteutuksen, mitoitukset ja hienosäädöt toteuttaa rakennusinsinööri.

2.1 Ohjelmistoarkkitehtuurin standardi ISO/IEC 42010

On olemassa useita standardeja, joiden mukaan ohjelmistoarkkitehtuureja voidaan suunnitella. Tässä kappaleessa käsitellään ISO/IEC standardia 42010. ISO on kansainvälinen standardointielin.

Standardi ei ole tarkoitettu sovellettavaksi ainoastaan ohjelmistotekniikkaan. Prosessijohtamisessa voidaan käyttää samoja malleja, joten standardi määrittelee erinäisiä arkkitehtuurimalleja. Jokaiselle arkkitehtuurimallille on omat rajaehdonsa, joiden tulee täytyä, kun haetaan kyseisen standardin hyväksyntää. Termien määrittämisen jälkeen standardissa esitetään malli, johon kaikki arkkitehtuurikuvaukset perustuvat. Standardiin pohjautuva malli on esitetty kuviossa 2.1.



Kuvio 2.1: Prosessijohtamisen rajapinnat ISO42010 standardin mukaisesti

Mallissa kaiken keskellä on jokin tietty ihmisen tekemä järjestelmä. Sen toiminnan yleiskuvan ja liitännäisyyden määrittää sen *de facto* arkkitehtuuri, joka taas pohjautuu arkkitehtuurikuvaukseen. Arkkitehtuurikuvauksella ei kuitenkaan tehtäisi yhtään mitään, jos järjestelmälle ei olisi jännitettä. Tarve uudelle mittausjär-

jestelmälle voi tulla joltain taholta, joka on valmis sijoittamaan siihen rahaa. Haak-sirikkoutuneelle seurueelle johtamisjärjestelmän ja toimivan yhteiskunnan luominen ovat luonnollisia tarpeita, jotka luovat jännitteen luoda ja ylläpitää sellaista järjestelmää. Arkkitehtuurin rattaat pyörivät jännitteen vaikutuksesta, mutta ne eivät pääse pakenemaan laakereiltaan: kaikki järjestelmät toimivat tietyissä ympäristöissä, joissa on aina tietyt rajatekijät ja muuttujat (ISO 42010, 3).

Standardissa käsitellään arkkitehtuurisuunnittelun osa-alueita ja rakennuslohkoja, sekä niiden välisiä relaatioita. Se kertoo filosofiaa, jonka mukaan arkkitehtuurisuunnittelua ja päätöksiä pitäisi tehdä. Arkkitehtuurin järkevyyden ja ongelmat joihin suunnittelupäätös liittyy vaikuttavat siihen, mikä ratkaisu pitää valita. (ISO 42010, 8)

Tämän opinnäytetyön olennainen osa on itse arkkitehtuurikuvaus. Standardi määrittelee, että arkkitehtuurikuvauksessa pitää tunnistaa kyseessä oleva järjestelmä (rajapinnat, jäsenet) osa-alueilta, jotka ovat fundamentalistisesti tärkeitä jännittäville tekijöille. Esimerkiksi seuraavia jännittäviä tekijöitä pitää harkita arkkitehtuurikuvausta laatiessa:

- käyttäjät,
- omistajat,
- tukijat,
- kehittäjät,
- rakentajat ja
- ylläpitäjät.

Arkkitehtuurikuvauksessa pitää myös miettiä seuraavia tekijöitä:

- järjestelmän tarkoitus,
- arkkitehtuurin sopivuus siihen, että järjestelmä onnistuu tavoitteissaan,
- koko järjestelmän suunnittelun ja toimeenpanon yleinen järkevyyden,
- potentiaaliset riskit jännittäville tekijöille elämänkaarensa aikana, sekä
- huollettavuus ja järjestelmän kehittyvyys (ISO 42010, 12).

Standardissa on kolme lisäystä, a, b ja c. Niissä käsitellään termistöä, nimeämisasioita sekä konsepteja (ISO 42010, 19-34).

3 AACT -järjestelmä

Johtuen vaiheistettujen antennien alati kasvavasta kysynnästä, nopea tapa mitata säteilukuviota tuotantoympäristössä on hyvin tarpeellinen antennivalmistajille. Järjestelmän pitää pystyä mittaamaan antenni järkevässä ajassa, sekä sen pitää olla kompaktin kokoinen.

Näillä ehdoilla tavanomainen säteilykuvion mittaaminen kaukokentästä on mahdotonta, sillä järjestelmällä pitää mitata antennin lähikenttää. Tämän lisäksi säteilykuvion mittaamiseen pitää käyttää antennimatriisia, jotta aikatavoite ei karkea käsistä.

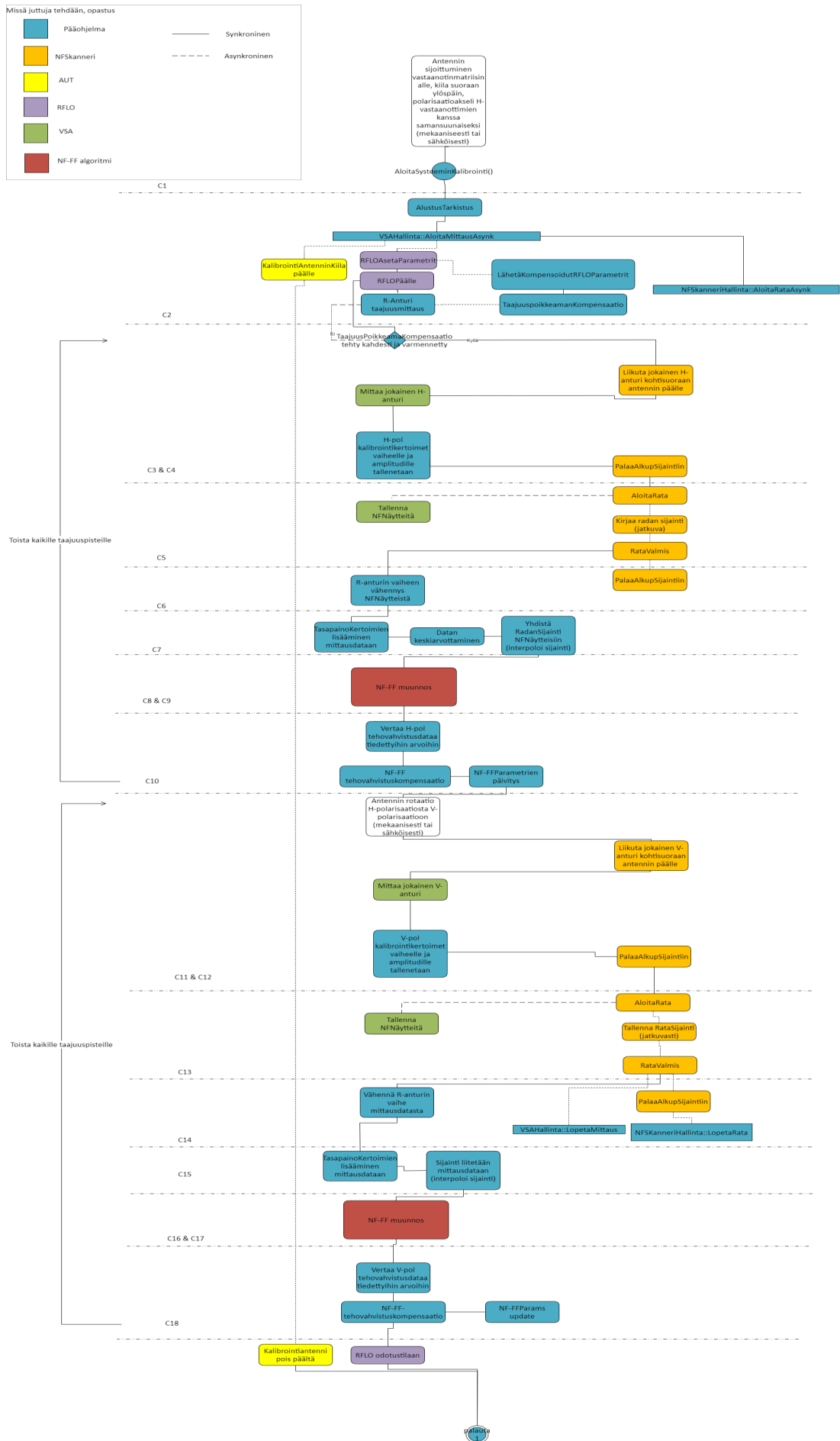
Tässä kappaleessa ei käsitellä mittausjärjestelmän laitekohtaisia, fyysisiä tai insinööriteknisiä seikkoja, sillä ne ovat ohjelmistoarkkitehtuurin rajojen ulkopuolella; ohjelmistoarkkitehtuuri on kuitenkin vain yleistason kuvaus järjestelmästä.

3.1 Järjestelmän kalibrointi

AACT -järjestelmä testaa ja kalibroi antenneja. Tämä tehtävä on hyvin monimutkainen. Järjestelmän toiminta koostuu pääpiirteittäin seuraavista asioista:

- järjestelmäkalibrointi (AACT:n kalibrointi mittauksia varten),
- antennin mittaus ja
- antennin kalibrointi.

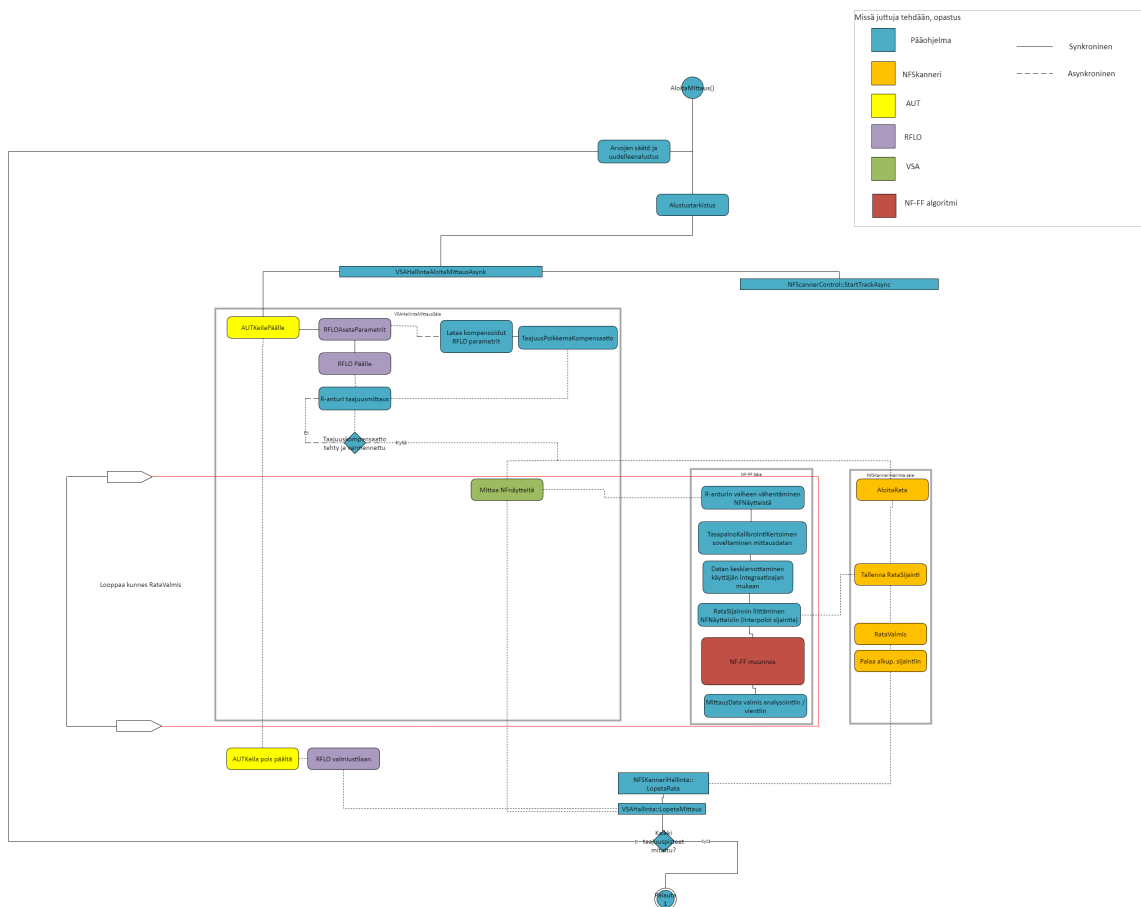
Mittauskojeisto kalibroidaan asettamalla paikalliskoskillaattorille oikea taajuus ("vaihelukittu" antennin taajuuteen), jonka jälkeen horisontaaliset ja pystysuuntaiset anturit asetetaan säteilevän antennin keskiakselille. Tästä saadaan kalibrointikertoimet jokaiselle anturille, joita käytetään lähikenttä-kaukokenttämuunnoksessa (NF-FF). Kun kertoimet ovat saatu, laitteisto ajaa radan antennin päällä, ja muunnos suoritetaan. Muunnetusta kaukokenttäkuvioista voidaan tehdä johtopäätös siitä, onko laitteen kalibraatio kunnossa. Kun kertoimet ovat optimaaliset, ne talletetaan pääohjelman muistiin, sillä niitä tarvitaan mittausdatan käsittelemisessä. Kuviossa 3.1 on esitetty AACT-järjestelmän kalibrointi kaaviona.



Kuvio 3.1: AACT -järjestelmän kalibrointi

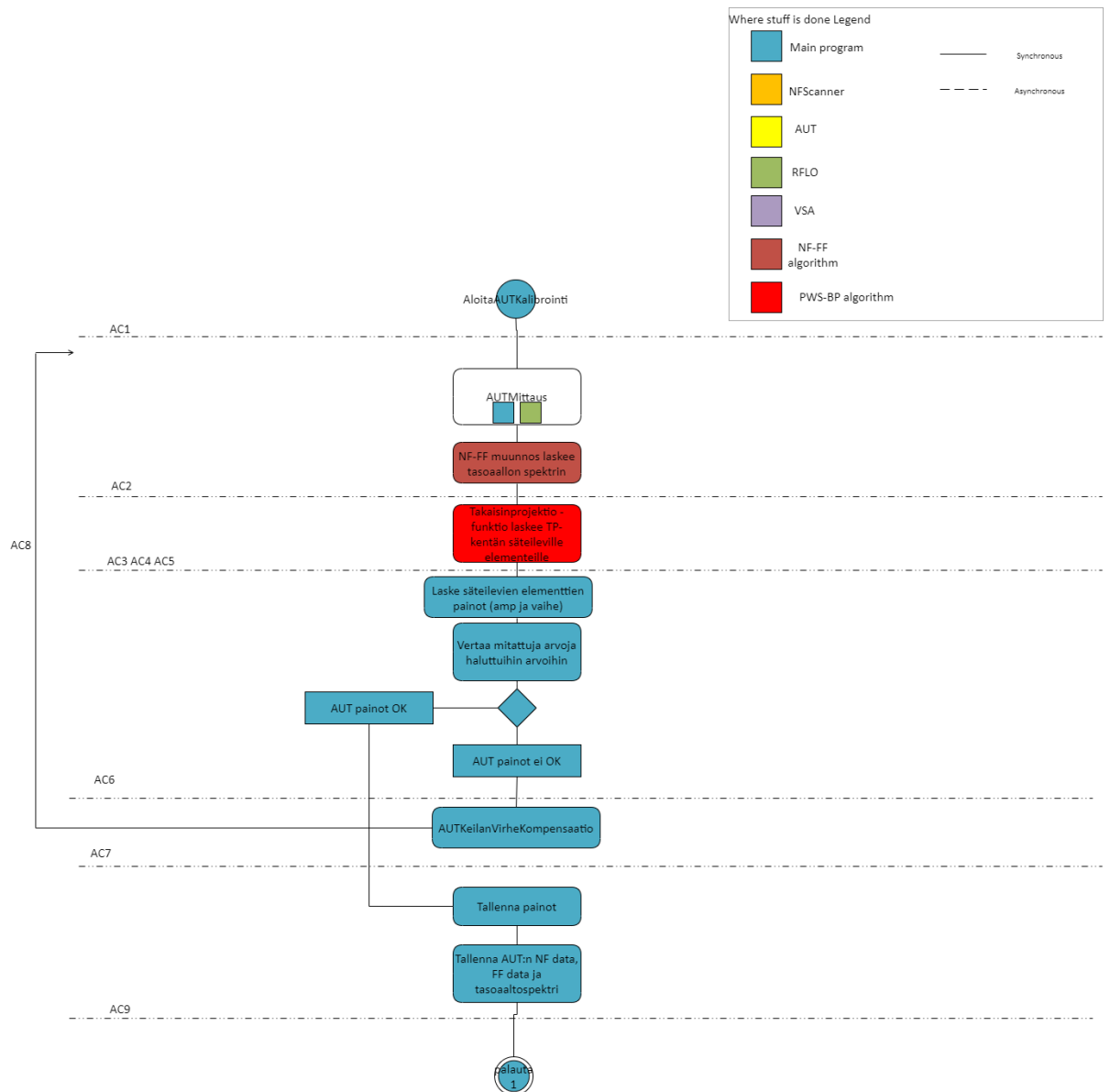
3.2 Antennin mittaus- ja kalibrointi

Antennin mittaaminen suoritetaan mittaamalla sen lähikentän säteilykuvio. Antenni asetetaan säteilemään määrättyyn suuntaan, ja lähikenttäskanneri ajaa käyttäjän määrittämän radan antennin päällä mitaten sen säteilykuviota. Lähikenttädataan liitetään aikaleima sekä sijaintitieto lähikenttäskannerilta (interpolatio tarvittaessa). Mittaamisen jälkeen data syötetään lähikenttä-kaukokenttämuunnos -algoritmiin, joka laskee säteilevän antennin kaukokenttäkuviota, tasoaaltospektrin, sekä takaprojektion. Takaprojektista nähdään säteilevän antennin painot (vaihe ja amplitudi) apertuurin pinnalla, josta voidaan havaita vialliset apertuurit. Pääohjelma laskee korjauskertoimet viallisille apertuureille. Laskennan jälkeen kertoimet ladataan antenniin. Lataamisen jälkeen mittaus toistetaan kunnollisen toiminnan varmentamiseksi. Alla olevassa kuviossa on esitetty kaavio antennin mittausproseduurista.



Kuvio 3.2: Antennin mittausproseduuri

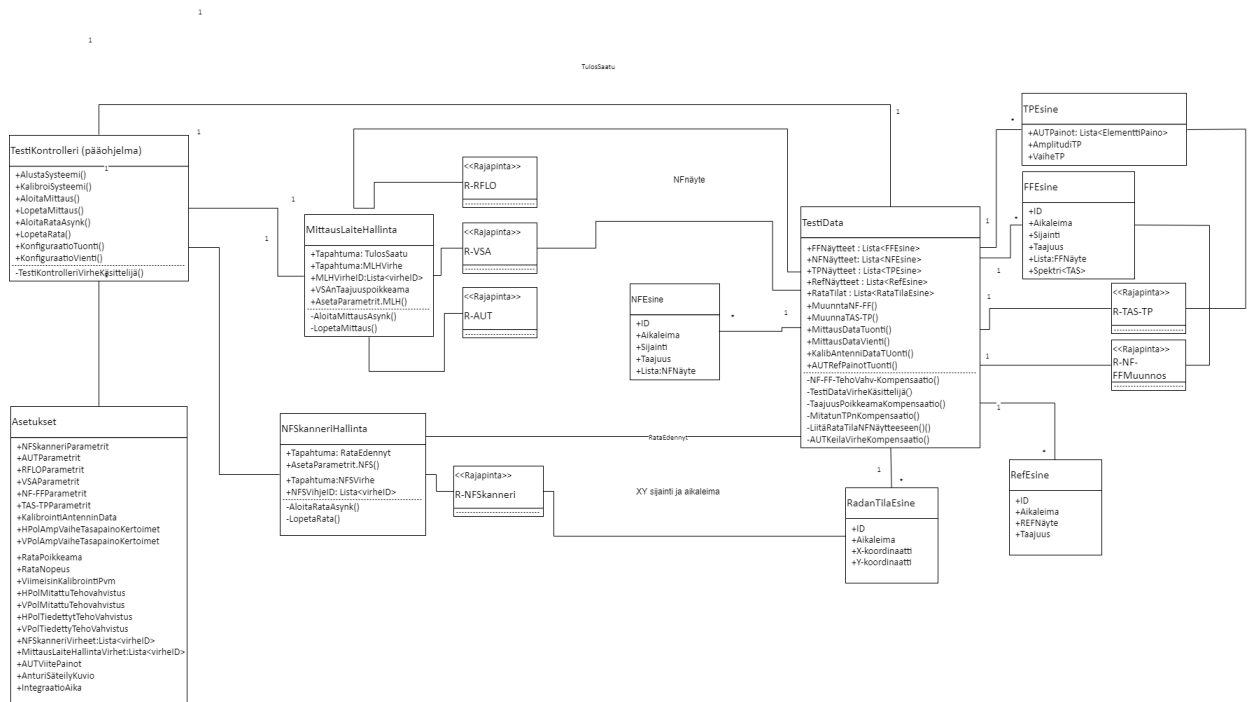
Kuviot 3.1 ja 3.2 on liitetty opinnäytetyöhön isommassa koossa. Lähi-kaukokenttämuunnos voidaan ajaa mittauksen aikana käyttäen datavirtaa. Käyttäjän halutessa niin, se voidaan myös ajaa vasta, kun mittaus on ajettu kokonaan loppuun. Jälkimmäinen tapa nopeuttaa laskentaa, mutta mittausproseduurissa kuluva kokonaisaika kasvaa. Kalibrointiproseduuri on havainnollistettu allaolevassa kuviossa, sekä esitetty isompana toisena liitteenä.



Kuvio 3.3: Antennin kalibrointiproseduuri

3.3 Data- ja hallintaluokat

Mittausjärjestelmässä liikkuu paljon dataa. Käyttäjän määrittämiä parametreja, sekä niistä määräytyviä vakioita käytetään pääohjelman toiminnassa, mittaamisessa sekä mittausdatan analysoinnissa. Kuvio 3.4 havainnollistaa data- ja hallintaluokat, sekä niiden väliset riippuvaisuudet ja datavirran.

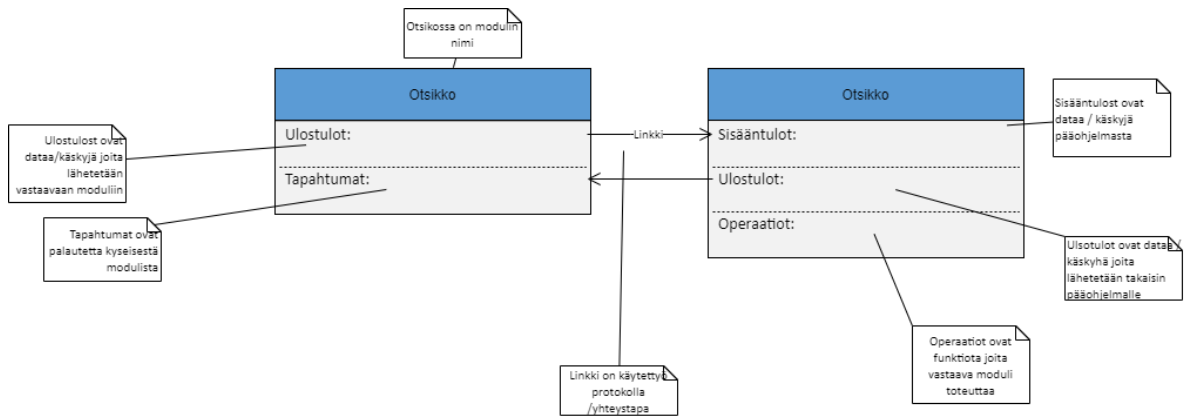


Kuvio 3.4: Data- ja hallintaluokat

Kuvio koostuu löyhästi kasatuista tietueista ja rajapinnoista. Viivat tietuitten välissä kuvastavat suoraa riippuvuutta kahden esineen välillä, esimerkiksi "Asetukset" -tietue on "TestiKontrolleri" -tietueen tietopankki. "TestiKontrolleri" -tietueessa on taas kuvattu funktionaalisuutta joissa tarvitaan kyseistä tietopankkia.

3.4 Rajapinnat

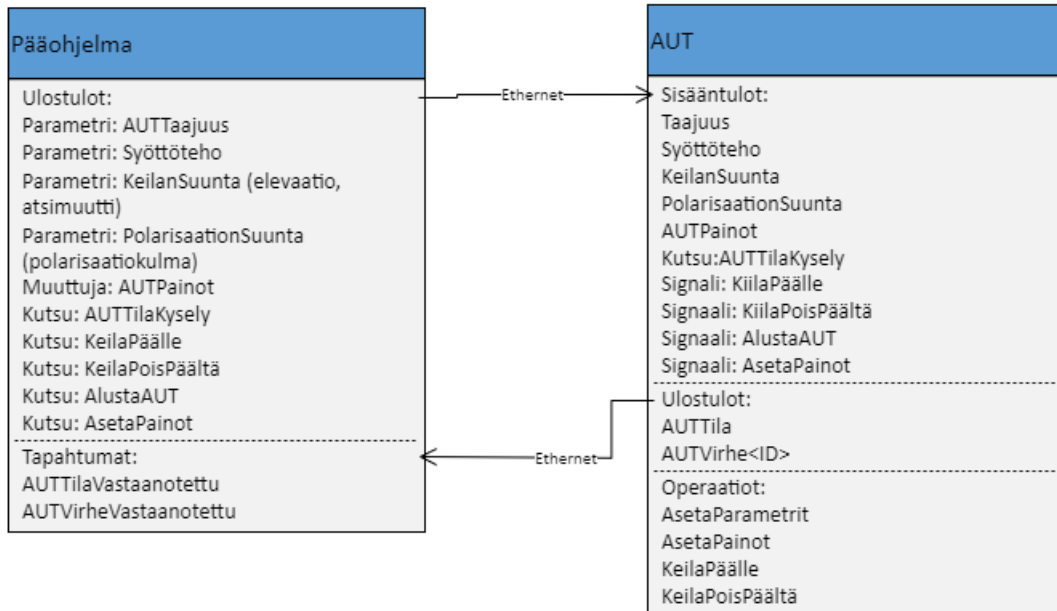
Kuvio 3.4 ei anna tyhjentävää kuvausta pääohjelman ja siihen kiinnitettyjen laitteisto- ja ohjelmistomodulien välillä. Tämä alikappale käsittelee tarkemmin eri modulien sisään- ja ulostuloja, sekä suoritettavia operaatioita. Kuvio 3.5 esittää rajapintakuvauksien tulkitsemisen ohjeistuksen.



Kuvio 3.5: Rajapintakuvauksien selite

Jokainen kuvaus tehtiin käyttämällä kyseistä mallia yhtenäisyyden säilyttämiseksi. Rajapintakuvaukset tehtiin vain suurimpien kokonaisuuksien välille, matalan tason liityntöjä ei ohjelmistoarkkitehtuurissa kuvata.

3.4.1 AUT

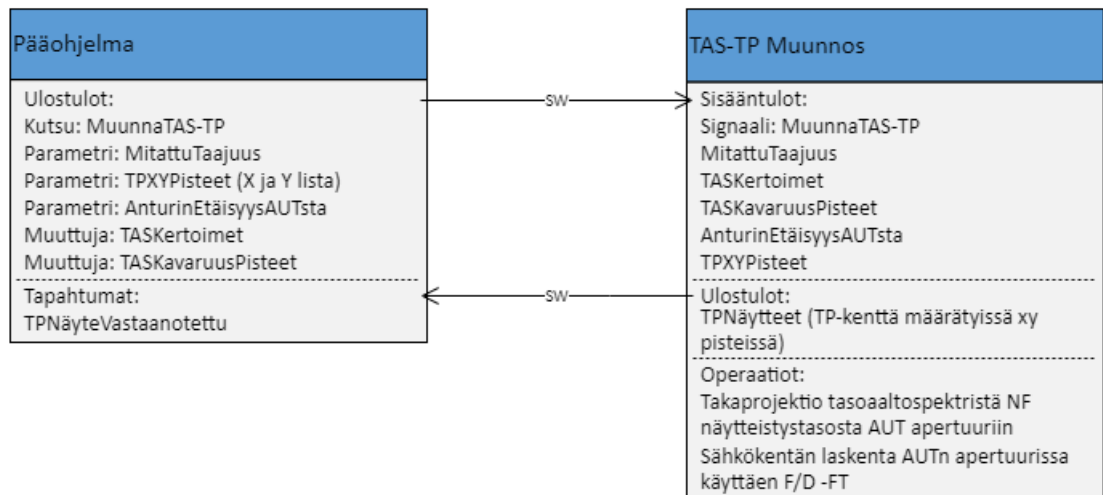


Kuvio 3.6: Pääohjelma – AUT -rajapinta

Testattavat antennit pitää saada lähettämään radioaaltoja käyttäjän määräämään suuntaan halutulla taajuudella. Keila pitää voida kytkeä sekä päälle että pois

päältä. Jotta laitteen pääfunktionaalisuus toteutuu, antennin painoja (apertuurien amplitudi ja vaihe) pitää myös pystyä säätämään.

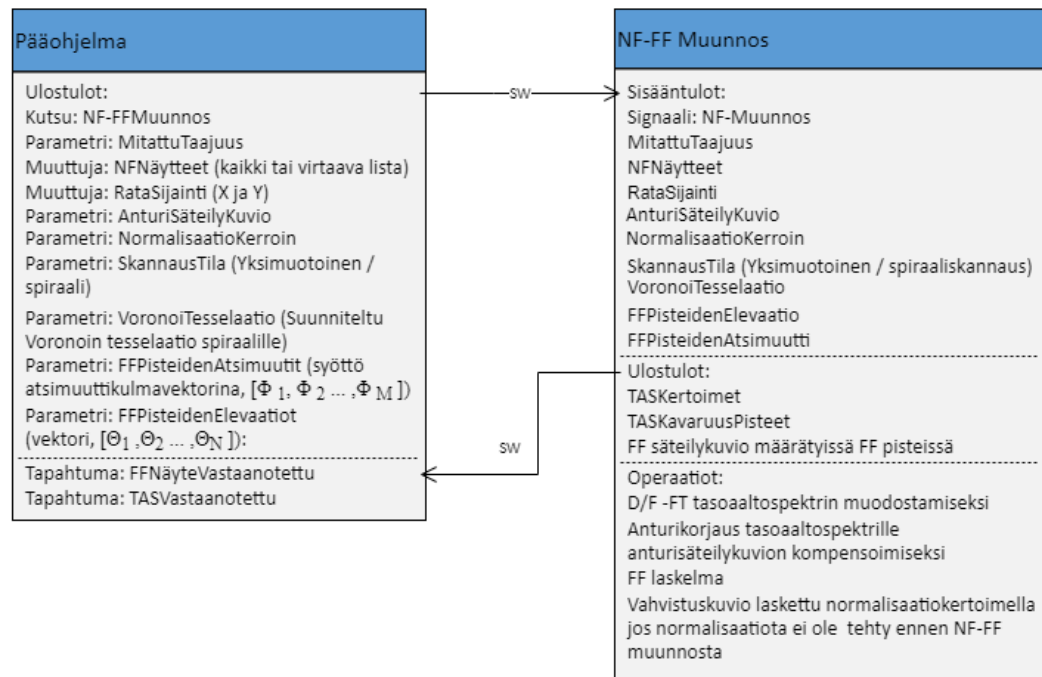
3.4.2 FF-TP



Kuvio 3.7: Kaukokenttä – takaprojektio -rajapinta

Järjestelmässä on kaksi isompaa ohjelmistokokonaisuutta pääohjelman lisäksi. Tasoaaltospektri-takaprojektioimuunnos on toinen niistä. Algoritmiin pitää syöttää lähikenttä-kaukokenttämuunnoksesta syntynyt tasoaaltospektri, joka muunnetaan takaprojektioiksi (hologrammi). Tästä hologrammista voidaan päätellä sähkökenttä antennin säteilevien apertuurien pinnalla. Sähköisesti ohjatut antennit toimivat apertuurien vaiheistuksella, joten hologrammin vaihetieto paljastaa apertuurit, joiden säteily eriaa optimista. Nämä apertuurit kalibroidaan (painoja muutetaan), jotta lopullinen säteilykuvio antaa parhaan mahdollisen suorituskyvyn.

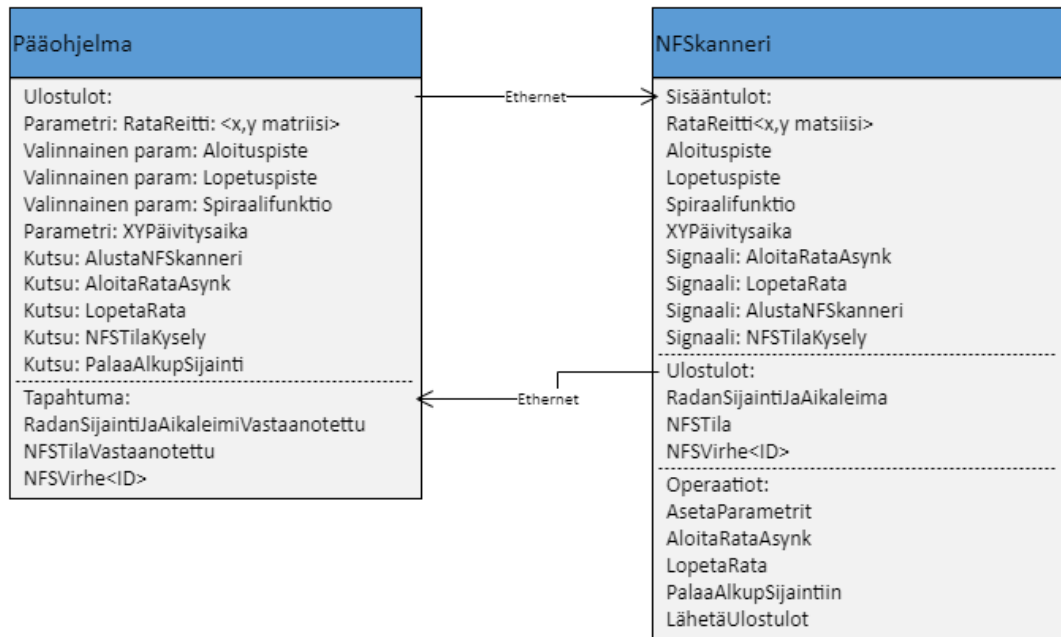
3.4.3 NF-FF



Kuvio 3.8: Pääohjelma – NF-FF -rajapinta

AACT-järjestelmä mittaa antennien lähikenttää, joka voidaan muuntaa kaukokentäksi matemaattisesti. Tämä muunnos toimii koko järjestelmän perustana, sillä kaukokentän mittaaminen on hyvin epäkäytännöllistä tuotantoympäristössä.

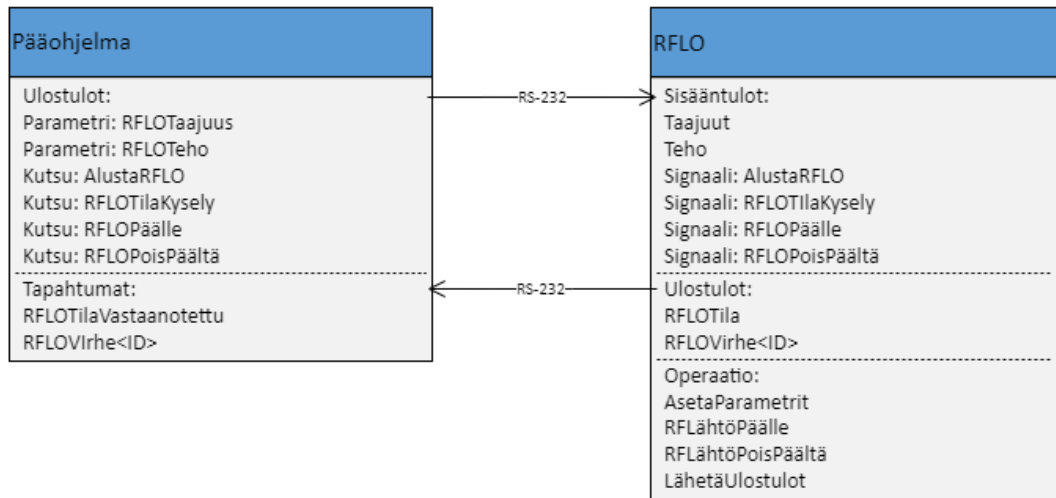
3.4.4 NF-skanneri



Kuvio 3.9: Pääohjelma – NFskanneri -rajapinta

Säteilykuvion mittaaminen vaatii sen, että sitä mitataan useista eri kohdista. Lähikenttäskannerin funktio on liikuttaa vastaanotinmatriisia säteilevän antennin päällä, antaen tarkan aika-sijaintitiedon pääohjelmalle. Skanneri voi ajaa useita eri ratakonfiguraatioita erilaisilla nopeuksilla. Skannerin toiminnan pitää myös olla tarkkaa, sillä mikroaaltoalueella aallonpituudet ovat pieniä, ja järjestelmän matematiikka vaatii hyvin pienen virhemarginaalin.

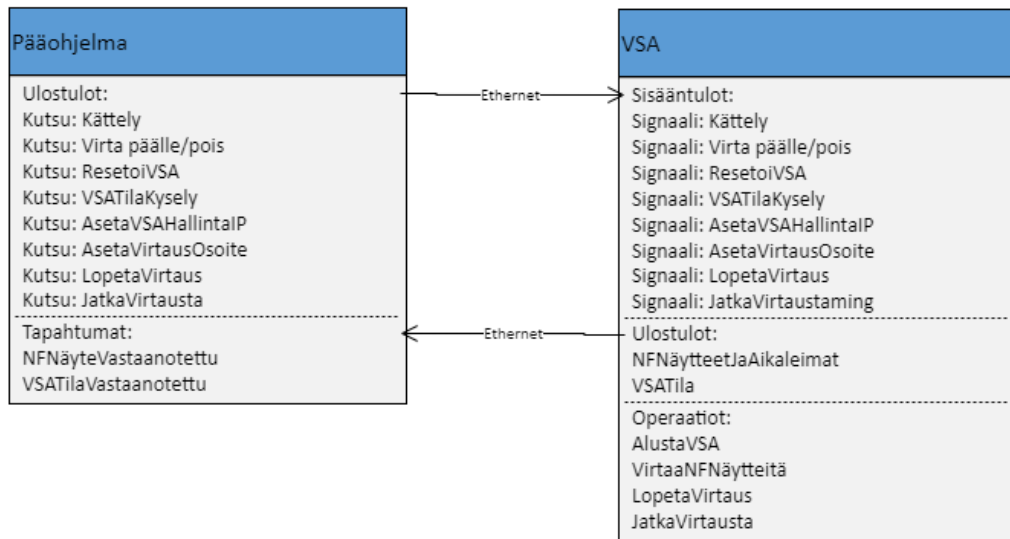
3.4.5 RFLO



Kuvio 3.10: Pääohjelma – RFLO -rajapinta

Mikroaaltojen nimi tulee siitä, että niiden aallonpituus on mikrometrien luokkaa. Aallonpituuden resiprokaali on taajuus. Suuritaajuiset signaalit ovat vaikeita käsitellä, joten ne pitää alassekoittaa pienemmälle taajuudelle. Paikallisoskillaattori on kiinni vastaanotinmatriisiketjun lopussa, jossa sekoitus pienemmälle taajuudelle tapahtuu. Oskillaattorin taajuuden pitää säätää modulaarisesti lähetettävän taajuuden mukana, sillä muuten sekoitus ei onnistu oikein.

3.4.6 VSA



Kuvio 3.11: Pääohjelma – VSA -rajapinta

Kun korkeataajuinen signaali on muunnettu pienemälle taajuudelle, sen tiedot pitää ottaa ylös. Monikanavaisen vastaanotinmatriisin jokaisen kanavan data muutetaan analogisesta digitaalisiksi IQ -näytteiksi VSA:ssa. Näitä näytteitä prosessoidaan lähikenttä-kaukokenttämuunnoksessa.

4 POHDINTA

Arkkitehtuurisuunnittelu on ohjelmistosuunnittelun lailla haastavaa työtä. Arkkitehdin pitää omaksua järjestelmän toiminta, sekä pilkkoa se osa-alueisiin. Sisään- ja ulostulot eri moduleille pitää miettiä, vaikkakin hyvin korkealla tasolla. Voisi jopa sanoa, että arkkitehtuurisuunnittelu muistuttaa enemmän järjestelmäsuunnittelua kuin ”koodaamista”, sillä korkean tason kuvaus ei määritä osoittimia, ohjelmakutsujen parametreja tai muita ohjelmistoteknisiä aspekteja.

Vanha sananlasku menee ”hyvin suunniteltu on puoliksi tehty”. Tämä pitää paikkansa arkkitehtuurisuunnittelun osalta, sillä hyvän arkkitehtuurisuunnittelun pohjalta on ohjelmistoinsinöörien helppo aloittaa työnsä. Projektiryhmällä on selvät rajaehdot sekä nimet asioille, joka auttaa työn selkeydessä, sekä vähentää sekaannuksista aiheutuvia ongelmia. Kuten monilla asioilla, tälläkin on kääntöpuolensa; huono arkkitehtuurisuunnittelu vaikeuttaa ohjelmoijien työtä, sillä epäloogisuudet ja epäjohdonmukaisuudet viivästävät työnvirtaa, puhumattakaan muista haitoista.

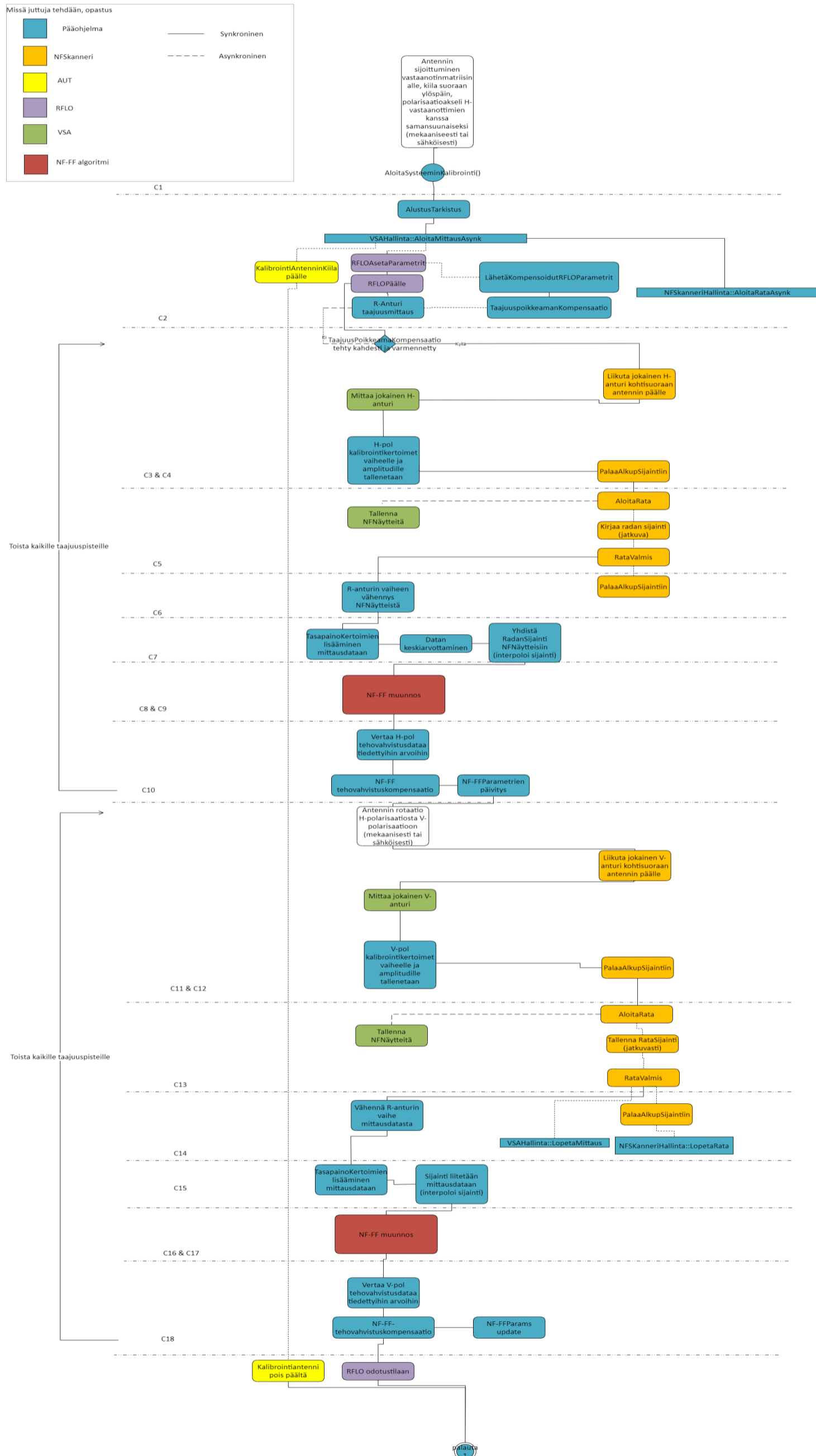
Tämän opinnäytetyön ohjelmistoarkkitehtuuri ei täytä ISO 42010 -standardia, eikä sitä rakennettukaan se mielessä. Vain luoja ja ohjelmistosuunnittelijat, jotka tulevat mahdollisesti käyttämään opinnäytetyön arkkitehtuuria, tulevat tietämään sen toimivuuden tai toimimattomuuden.

LÄHTEET

ISO42010, Systems and software engineering — Architecture description, Luettu 6.9.2020. <https://www.iso.org/home.html>

LIITTEET

Liite 1. Järjestelmäkalkibroiinti -täysikokoinen



Liite 2. Antennin mittausproseduuri -täysikokoinen

