

Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Sulautetut ohjelmistot

2016

Kimmo Kummala

ULKOISEN TIEDON KÄYTTÄMINEN SÄÄTÖJÄRJESTELMÄSSÄ

– Automaatiojärjestelmän optimointi

Kimmo Kummala

ULKOISEN TIEDON KÄYTTÖ SÄÄTÖJÄRJESTELMÄSSÄ

- Automaatiojärjestelmän optimointi

Tämän työn tarkoituksena oli tutkia ja toteuttaa Timil Oy:lle talon säätöjärjestelmän logiikkaan ohjelma, joka ulkoista palvelua käyttämällä hakee sääennusteen seuraavalle 36 tunnille. Työssä piti selvittää tähän tarkoitukseen soveltuva ulkoinen palvelu ja sieltä saatavilla olevat tiedot. Lisäksi haluttiin tutkia mahdollisuutta käyttää ulkopuolista palvelua seuraavan päivän sähköhintatiedon saamiseksi säätöjärjestelmän käyttöön. Ohjelma ja sen käyttämät palvelut mahdollistavat koko järjestelmän säätämisen etukäteen tulevan säätilan ja sähköhinnan perusteella.

Sääennusteen saamiseksi käytettiin Ilmatieteen laitoksen Avoin Data -rajapintaa. Ensin tutkittiin, miten tämä rajapinta toimii ja millä hakukriteereillä sieltä saadaan oikeanlaista tietoa. Tämän jälkeen talon säätöjärjestelmään toteutettiin sääennusteen hakemiseen tarkoitettu aliohjelma käyttämällä CODESYS-ohjelmointiympäristöä ja WAGO 775-881 -logiikkakontrolleria.

Seuraavan päivän sähköhinta voidaan hakea käyttämällä Nord Poolin tarjoamaa Elspot FTP -palvelua, joka on maksullinen palvelu. Tässä työssä tutkittiin, missä muodossa tieto on palvelussa, miten tieto saadaan palvelusta haettua ja mitä tämä palvelu maksaa. Samalla suunniteltiin se, millainen ohjelma WAGO:n kontrolleriin pitää tehdä, jotta sähkönn hinnat saadaan haettua palvelusta.

Sääennustetietojen hakemiseksi tehty aliohjelma täytti sille asetetut tavoitteet, ja työssä selvisi myös sähköhinnan hakemiseksi tarvittavat asiat.

ASIASANAT:

PLC, CODESYS, WAGO, Avoin data, Ilmatieteen laitos, Nord Pool, ELSPOT

Kimmo Kummala

USING EXTERNAL DATA IN CONTROLLING SYSTEMS

- Optimization of automation system

The goal of this thesis work was to study and implement the logic control system, which is using an external service by searching the weather forecast for the next coming 36 hours. Whole home automation control system is developed by Timil Oy and this thesis work covers one sub-component to that system. The main requirement was to find a suitable external weather forecast service provider and make a program which is requesting and analyzing data by using that service provider. Another minor requirement was to find out external service provider for the next day electricity price. Requirements was not to implement the program for searching next day electricity, but find out how to build this kind of system in the future.

The software program in this thesis work makes it possible to adjust whole system in advance based on next day weather forecast and the price of electricity.

The weather forecast was obtained by using the Meteorological Institution Open Data interface. At this work, it was studied how this interface works and what are the right search criteria's we need to use to get correct weather forecast data. A sub-program was made to get weather forecast data from the Open Data service by using the CODESYS programming environment and WAGO 775-881 logic controller.

The price of next day electricity can be searched by using Nord Pool Elspot FTP service. Companies have to pay when using the Nord Pool service. In this work, it was studied in which format the data is in the FTP service, how to get the data from the service, and what does this service cost. At the same time, it was studied what kind of program we need to make for WAGO controller in order to get next day electricity price from the FTP service.

The program for weather forecast searching met all the objectives set for it. You can search the weather forecast from the service, analyze the data and provide the needed data for the main program in the system.

The program for the next day electricity price is not implemented in this work. All objectives were met like the service for that purpose, how to get the data and the price and what are the needed software components for implementation.

KEYWORDS:

PLC, CODESYS, WAGO, Open Data, Ilmatieteen laitos, Nord Pool, ELSPOT

SISÄLTÖ

| | |
|--|-----------|
| KÄYTETYT LYHENTEET | 6 |
| 1 JOHDANTO | 7 |
| 2 KÄYTETYT TEKNIIKAT JA PALVELUT | 8 |
| 2.1 Käytetyt ohjelmointikielet | 8 |
| 2.2 PLC (ohjelmitava logiikka) | 10 |
| 2.3 Codesys-ohjelmointiympäristö | 10 |
| 2.4 WAGO-logiikkapiiri | 12 |
| 2.5 Avoin data Suomessa | 13 |
| 2.6 Ilmatieteen laitos avoin data | 13 |
| 2.6.1 Saatavissa olevat sääennustetiedot | 15 |
| 2.6.2 Tiedon hakeminen palvelusta | 15 |
| 2.6.3 Palvelusta saatava vastaus | 16 |
| 2.7 Nord Pool | 16 |
| 2.7.1 Spot-markkinat | 17 |
| 2.7.2 Spot-hinnan määräytyminen | 17 |
| 2.7.3 Spot-hintatietoahan kustannukset | 17 |
| 3 OHJELMAN SUUNNITTELU JA OHJELMOINTI | 20 |
| 3.1 Ongelmat, joita lähdettiin ratkaisemaan | 20 |
| 3.2 Arkkitehtuuri | 21 |
| 3.3 Sää-aliohjelma | 23 |
| 3.3.1 Aliohjelman parametrit | 23 |
| 3.3.2 Tilakone | 23 |
| 3.3.3 Haun tekeminen | 24 |
| 3.3.4 Vastauksen purkaminen | 25 |
| 3.4 Sähkön hinnan hakeminen | 25 |
| 3.5 Tietoliikenne | 26 |
| 3.5.1 Käytetyt verkkoprotokollat | 26 |
| 3.5.2 Tietoturva | 26 |
| 3.6 Testaus | 26 |
| 4 PÄÄTELMÄT | 28 |

LIITTEET

Liite 1. Ilmatieteen laitoksen palvelimelta tullut XML-tiedosto

KUVAT

| | |
|--|----|
| Kuva 1. Esimerkki Function Block Diagramista. | 8 |
| Kuva 2. Esimerkki Ladder Diagramista. | 9 |
| Kuva 3. Esimerkki Structure Textistä. | 9 |
| Kuva 4. Esimerkki Instruction List -ohjelmointikielestä. | 9 |
| Kuva 5. Esimerkki Sequential Function Chart -ohjelmointikielestä. | 10 |
| Kuva 6. WAGO 775-881 ohjelmitava Ethernet-kontrolleri. | 12 |
| Kuva 7. Esimerkki Ilmatieteen laitoksen palvelimelta tulevasta XML-tiedoston muodosta. | 16 |
| Kuva 8. Nord Poolin Elspot-hinnat xls-muodossa. | 18 |
| Kuva 9. Nord Poolin Elspot-hinnat sdv-muodossa. | 18 |
| Kuva 10. Esimerkki hinnoista Nord Poolin verkkosivuilla. | 19 |
| Kuva 11. Korkean tason arkkitehtuuri. | 21 |
| Kuva 12. Vuokaavio ohjelman toiminnasta. | 22 |
| Kuva 13. Ilmatieteen laitokselta haettu lämpötilaennuste. | 22 |
| Kuva 14. Aliohjelman sisäänmenevät ja ulostulevat parametrit. | 23 |
| Kuva 15. Sää-aliohjelman tilakone. | 23 |
| Kuva 16. Koodilistaus, tilakoneen alku. | 24 |

KÄYTETYT LYHENTEET

| | |
|------------|---|
| PLC | Programmable Logic Controller, ohjelmitava logiikka. |
| IEC | International Electrotechnical Commission, kansainvälinen sähköalan standardointiorganisaatio. |
| IEC 61508 | Teollisuusautomaation Toiminnallisen turvallisuuden standardi - https://en.wikipedia.org/wiki/IEC_61508 . |
| IEC61131-3 | Kansainvälinen standardi ohjelmitaville logiikoille, joka määrittelee muun muassa ohjelmointikielät. |
| OEM | Original Equipment Manufacturer, alkuperäinen laitevalmistaja. |
| CNC | Computerized Numerical Control, tietokoneistettu numeerinen ohjaus. |
| SIL2, SIL3 | Safety Integrity Level, suomeksi TET eli Turvallisuuden Eheyystaso. Tämä on turvallisuuteen liittyvälle sähköiselle ohjausjärjestelmälle eheyden vaatimusten määrittelyä luotu laskentatapa. |
| HIRLAM | High Resolution Limited Area Model, numeerinen sääennustemalli. |
| OGC | Open Geospatial Consortium (http://www.opengeospatial.org/standards). |
| WFS | Web Feature Service, rajapintastandardi maantieteellisten tietojen käsittelemiseksi internetin yli käyttäen alusta riippumattomia kyselyjä. |
| INSPIRE | Infrastructure for Spatial Information in the European Community, EU-direktiivi, jonka avulla luodaan EU:n jäsenmaiden yhteinen, yhtenäinen ja helposti hyödynnettävä paikkatietoinfrastruktuuri. |
| GPS | Global Positioning System, maailmanlaajuinen satelliittipaikannusjärjestelmä. |
| SDK | A Software Development Kit, työkalut ohjelmistojen tekemiseksi. |

1 JOHDANTO

Tämän työn tarkoituksena on tehdä talon säätöjärjestelmän ohjelmoitavaan logiikkaan aliohjelma, joka hakee säätiedot tulevalle 36 tunnille käyttämällä ulkoista palvelua. Lisäksi tutkitaan mahdollisuutta käyttää ulkopuolista dataa reaaliaikaisen sähkönhintatiedon saamiseksi kokonaisjärjestelmän käyttöön.

Työ tehdään Timil Oy:lle. Timil Oy on Liedon Tarvasjoella toimiva pieni perheyrittäjä, jonka tavoite on kehittää energiatehokkaita ja käyttäjäystävällisiä kotiautomaatiojärjestelmiä niiden käyttäjille. Järjestelmä valvoo rakennuksen eri parametreja, kuten lämpötilaa, kosteutta jne. ja tekee niiden avulla automaattiset säädöt järjestelmälle. Järjestelmät sisältävät muun muassa käyttöliittymän, ohjauksen ja säädöt kodin ilmastointikoneelle, maalämpöpumpulle ja muille siihen kytketyille laitteille.

Tehtävän ohjelman tarkoituksena on mahdollistaa järjestelmän säätäminen ennakoitavasti tulevan säätilan mukaan. Lisätoivomuksena on myös käyttää tulevan päivän sähkön pörssihintatietoa, jotta lämmitys tai muu sähköä kuluttava toimenpide voidaan tehdä mahdollisimman edulliseen aikaan. Ohjelma tulee olemaan aliohjelma isommassa järjestelmässä. Tämä ohjelma ei tee laitteiden lopullista päätöstä siitä, miten järjestelmää tullaan säätämään, vaan ohjelma antaa järjestelmälle lisää parametreja järjestelmän tarkalle ja tehokkaalle säätämiseksi.

Työssä käydään läpi projektiin liittyvät työkalut ja käytetyt menetelmät sekä selitetään ohjelman perusperiaate.

Ilmatieteen laitoksen Avoin data -palvelua on käsitelty muutamassa muussakin lopputyössä. [1], [2], [3]

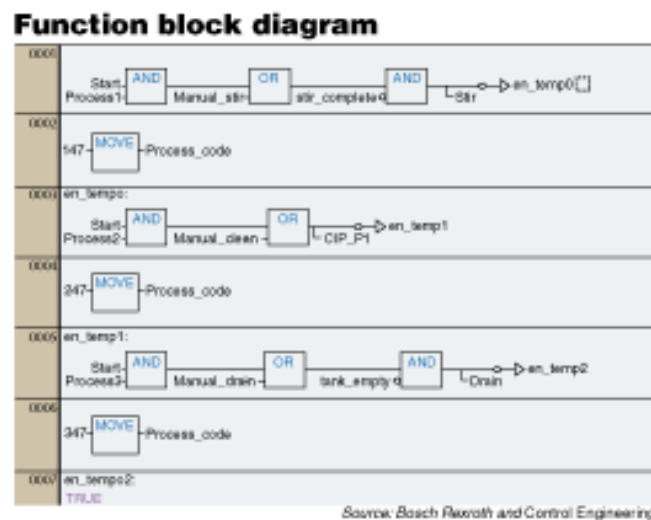
2 KÄYTETYT TEKNIIKAT JA PALVELUT

Työssä käytettiin monia erilaisia tekniikoita, välineitä ja palveluja. Tärkeimmät välineet olivat CODESYS-ohjelmointiympäristö WAGOn logiikkapiirille ja WAGOn logiikkapiiri. Tärkeimpiä palveluja olivat Ilmatieteen laitoksen avoimen datan palvelu sekä Nord Pool -sähköpörssi.

2.1 Käytetyt ohjelmointikiel

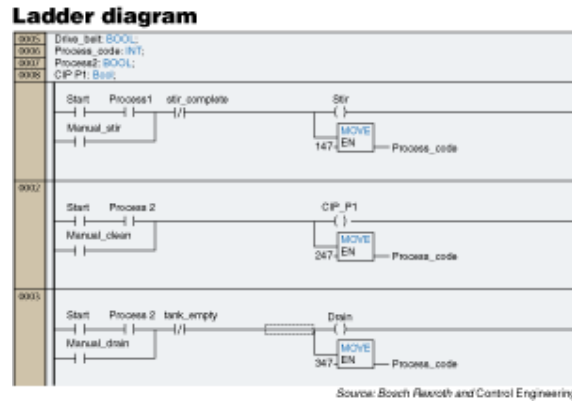
PLC-ohjelmointikielten perustana on kansainvälinen IEC 61131-3 -standardi. Se määrittelee syntaksit, semantics ja display viidelle standardi PLC-ohjelmointikielille, joita ovat [4]:

- FBD (Function Block Diagram)



Kuva 1. Esimerkki Function Block Diagramista.

- LD (Ladder Diagram)



Kuva 2. Esimerkki Ladder Diagramista.

- ST (Structure Text)

Structured text

```

0001 Process1: BOOL;
0002 Drive_Dat: INT;
0003 Process2: SOG1;
0004 CIP_P1: Bset;
0005
0006 IF (Process1 = TRUE AND start = TRUE OR manual_stir = TRUE) AND stir_complete = FALSE THEN
0007   stir := TRUE;
0008   Process_code := 147;
0009 END_IF;
0010
0011 IF (Process2 = TRUE AND start = TRUE OR manual_clean = TRUE) THEN
0012   CIP_P1 := TRUE;
0013   Process_code := 247;
0014 END_IF;
0015
0016 IF (Process3 = TRUE AND start = TRUE OR manual_drain = TRUE) AND tank_empty = FALSE THEN
0017   Drain := TRUE;
0018   Process_code := 347;
0019 END_IF;
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040

```

Source: Bosch Rexroth and Control Engineering

Kuva 3. Esimerkki Structure Textistä.

- IL (Instruction List)

Instruction list

```

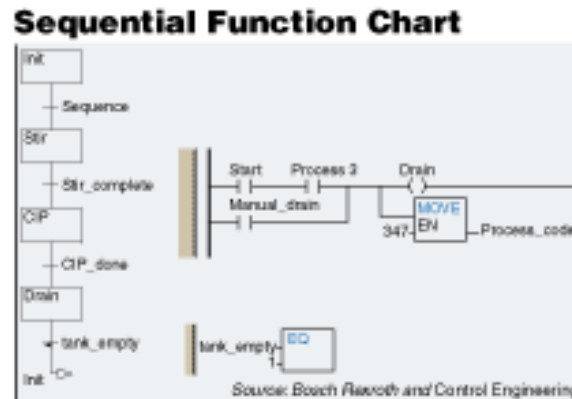
0001 LD Start
0002 AND Process1
0003 OR Manual_stir
0004 ANDN stir_complete
0005 ST Stir
0006
0007 JMPN en_temp0
0008
0009 LD 147
0010 MOVE
0011 ST Process_code
0012
0013 en_temp0:
0014 LD start
0015 AND Process2
0016 OR Manual_clean
0017 ST CIP_P1
0018
0019 JMPN en_temp1
0020
0021 LD 247
0022 MOVE
0023 ST Process_code
0024
0025 en_temp1:
0026 LD start
0027 AND Process3
0028 OR Manual_drain
0029 ANDN tank_empty
0030 ST Drain
0031
0032 JMPN en_temp2
0033
0034 LD 347
0035 MOVE
0036 ST Process_code
0037
0038 en_temp2:
0039
0040

```

Source: Bosch Rexroth and Control Engineering

Kuva 4. Esimerkki Instruction List -ohjelmointikielestä.

- SFC (Sequential Function Chart)



Kuva 5. Esimerkki Sequential Function Chart -ohjelmointikielestä.

Ohjelmoija voi valita käytettävän ohjelmointikielen sen mukaan, oman osaamisensa, ongelman, ulkoisten komponenttien tai esimerkiksi käytettyjen rajapintojen mukaan.

2.2 PLC (ohjelmoitava logiikka)

Ohjelmoitava logiikka on pieni mikroprosessoripohjainen tietokone tai laite, jota käytetään automaatioprosessien ohjaukseen. Itse ohjelma on patterivarmennetussa muistissa. Laitteessa on modulaarisia tai integroituja tulo- ja lähtöportteja, joihin on kytketty esimerkiksi verkkoliityntä, lämpötila-antureja ja moottorin käynnistimiä, merkkivaloja ja venttiilejä. Logiikka vaatii toimiakseen virtalähteen. Peruspakettiin kuuluvat yleensä kontrolleri ja digitaaliset sekä analogiset input- ja output-kortit. Tyypillisesti vaadittu käyttöjännite on 24V. Nykyaikaisella yhdellä ohjelmoitavalla PLC-logiikalla voidaan korvata kokonaisien relekeskusten kaikki ohjeukseen tarvittut komponentit. [5]

Tässä työssä käytetyllä ohjelmoitavalla logiikalla ohjataan kotiautomaatiojärjestelmään kytkettyjä laitteita, kuten ilmastointikonetta, venttiilejä, maalämpöpumppua jne.

2.3 Codesys-ohjelmointiympäristö

CODESYS on kokonaisvaltainen ohjelmointityökalu ja laitteistoriippumaton IEC 61131-3 -standardin mukainen ohjelmointiympäristö automaatio-, liikkeenohjaus- ja logiikkasovelluksiin. Ohjelmistotyökalun on valmistanut 3S-Smart Software Solutions GmbH ja se on johtava laitteistoriippumaton ohjelmisto Windows- tai Linux-pohjaisten laitteiden sekä sulautettujen ja ilman käyttöjärjestelmää olevien laitteiden ja sovellusten ohjelmointiin. [6]

CODESYS tuotteita ovat muun muassa: [6]

- CODESYS Engineering

Lisenssivapaa IEC 61131-3 -pohjainen ohjelmointiympäristö, joka sisältää kaiken tarvittavan ohjelmistojen tekemiseksi. Näitä ovat mm. editorit, viainentätyökalut ja kääntäjät. Kääntäjä tukee kaikkia yleisesti käytössä olevia prosessoreja, kuten ARM/Cortex, X86/PentiumX, PowerArchitecture ja TriCore. Ohjelmistoa voidaan laajentaa erilaisilla liitännäisillä ja ohjelmilla, joilla saadaan tuki mm. korkeamman tason ohjelmointikielille (C, C++, etc.).

- CODESYS Runtime

- Laitevalmistajille (OEM) tarkoitettu SoftPLC Runtime System. Jotta laite olisi ohjelmitavissa CODESYS-ohjelmilla, siihen pitää asentaa CODESYS Runtime System. Tämä ajoympäristö tekee mistä tahansa sulautetusta tai PC-pohjaisesta laitteesta IEC61131-3- ohjauslaitteen, joka on ohjelmitavissa CODESYS-ohjelmilla.

- CODESYS Visualization

- Pitää sisällään kolme erilaista IEC61131-3 -käyttöliittymä vaihtoehtoa. CODESYS TargetVisu toimii samassa ohjaimessa yhdessä CODESYS Control RTE/Win kanssa. Web-palvelimen kanssa kommunikoiva CODESYS WebVisu on HTML5-kielellä toteutettu käyttöliittymä, joka toimii nettiselaimella. CODESYS HMI on käyttöliittymä ilman PLC-toimintoja. Sillä voidaan tehdä liityntä eri alustoille kuten Windows XP, Win CE ja Linux.

- CODESYS Fieldbus

- CODESYS Motion + CNC

- Tämä ohjelmisto on tarkoitettu vaativiin liikkeenohjausohjelmistoihin, kuten CNC ja CAM. Alustana käytetään Windows käyttöjärjestelmällä CODESYS Control RTE ja Linux käyttöjärjestelmällä CODESYS Control -ohjelmistoja.

- CODESYS Safety

- Tämä yhdistää turvalogiikan ohjelmoinnin samaan ohjelmointiympäristöön muun CODESYS-ohjelmoinnin kanssa ja mahdollistaa toiminnallisen turvallisuuden SIL2- ja SIL3 -eheyttasot standardin IEC 61508 mukaan.

- CODESYS PLCHandler

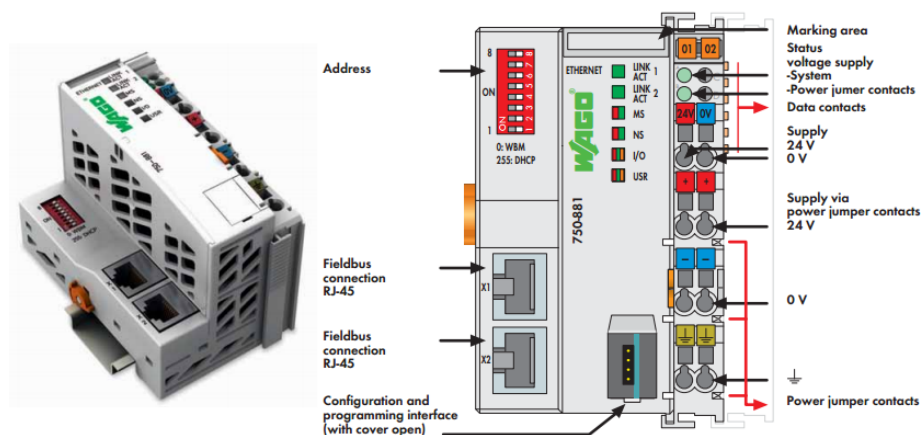
- SDK, jolla on helppo luoda ohjelmointirajapinta. SDK mahdollistaa tiedon- siirron kolmannen osapuolen laitteiden kanssa ja sen voi liittää erilaisille järjestelmälustoille.

Tällä hetkellä yli 500 OEM-laitevalmistajaa käyttää CODESYS-ohjelmistoa. Vuonna 2013 laitelisenssejä toimitettiin yli 500 000 laitteeseen eri teollisuudenaloille [6].

2.4 WAGO-logiikkapiiri

WAGO on ruuvittoman, jousivoimaisen liitinteknologian markkinajohtaja ja yksi johtavista tähän teknologiaan liittyvien laitteiden ja automaatiojärjestelmien valmistajista. Yhtiön laaja tuotevalikoima kattaa riviliittimet, liittimet, PCB-liittimet, elektroniset moduulit ja I/O-järjestelmät, joista jokainen tarjoaa teollisuusautomaatio- ja rakennusinstallaatioasiak- kaille valtavia etuja. [7]

WAGO Kontakttechnik GmbH & Co valmistaa tässä työssä käytettyä ohjelmoitavaa Et- hernet logiikkakontrolleria 775-881. Se toimii 24 voltilla, siinä on tosiaikakello ja siihen on mahdollista kytkeä 2 RJ-45 kaapelia. Piiri tukee FTP-, http-, SNMP-, DHCP-, DNS-, BootP-, ModbusTCP-, Modbus UDP- ja ETHERNET-IP-protokollia. CPU on 32-bittinen. Sisäistä ohjelmamuistia on 1024 kB ja datamuistia 512kB. Datapuskurissa on kestopuist- tia 32kB ja 2MB Flash-muisti. Logiikkapiirin hinta tätä työtä tehdessä oli noin 500 euroa. [8]



Kuva 6. WAGO 775-881 ohjelmoitava Ethernet-kontrolleri.

2.5 Avoin data Suomessa

Tieto on pääomaa, jonka arvo kasvaa sitä jaettaessa ja jalostettaessa. Julkishallinnolla on hallussaan huomattavat tietovarannot. Helpottamalla tietovarastojen käyttöä voidaan lisätä tämän tietopääoman arvoa. Samalla voidaan luoda mahdollisuuksia uudelle liiketoiminnalle, tutkimukselle ja koulutukselle. Lisäksi tietovaranto avulla voidaan kehittää myös parempia palveluja kansalaisille, parantaa päätöksentekoa, tehostaa hallinnon toimintaa ja lisätä sen läpinäkyvyyttä. [9]

Avoin julkinen data on julkisen organisaation tuottamaa tai hallinnoimaa tietoa, joka on konekielisessä muodossa ja maksutta kenen tahansa käytettävissä, muokattavissa ja jaettavissa sekä yksityisiin että kaupallisiin tarkoituksiin. Tietovarantojen avaamista on vuosina 2013–2015 vauhditettu avoimen tiedon ohjelman toimenpitein. [9]

Suomessa on avattu merkittäviä tietovarantoja, kuten maastotietoja, sää tietoa, liikenne- ja ajoneuvotietoa, tilastoja, talousdataa ja kulttuuriaineistoja. Avoimen tiedon ohjelma 2013–2015 käynnistettiin keväällä 2013 vauhdittamaan ja koordinoimaan julkisen hallinnon tietovarantojen avaamista. Ohjelman puitteissa on toteutettu muun muassa avoimen tiedon ja yhteistoimivuuden palvelu Avoindata.fi ja avointa dataa hyödyntävä kokeiluympäristö JulkICTLab. [10]

2.6 Ilmatieteen laitos avoin data

Tässä työssä haluttiin tutkia, miten Ilmatieteen laitoksen Avoin data -palvelusta saadaan selville paikkakunta-kohtainen sääennuste.

Ilmatieteen laitos on kehittänyt avoimen datan palvelun vuosina 2007–2013. Verkkopalvelu avattiin virallisesti 15.5.2013. Verkkopalvelu tarjoaa Ilmatieteen laitoksen tietoaineistoa ja dataa maksutta julkiseen käyttöön. Palvelun kautta voi hakea, katsella ja ladata tietoaineistoa digitaalisessa koneluettavassa muodossa (esim. XML). [11]

Avattavat tietoaineistot voidaan jakaa kolmeen eri luokkaan:

- sää-, meri- ja ilmastohavainnot
- tutka- ja salamahavainnot
- kansallisen sääennustemallin tiedot

Tietoaineisto jaetaan kolmeen eri kategoriaan: [11]

1. Reaaliaikaiset tiedot
 - a. Asemakohtaiset havainnot (esim. tuuli-, lämpötila-, kosteus-, ilmanpaine-, sade-, merivedenkorkeus- ja aallokkohavainnot)
 - b. Sääatutkakuvia ja salamoiden paikannustiedot Suomen alueelta
2. Aikasarjat
 - a. Ilmastohavainnot vuodesta 1959 (asemakohtaisia päivä- ja kuukausiarvoja)
 - b. Meriveden korkeushavainnot vuodesta 1971
 - c. Aaltohavainnot vuodesta 2005
3. Ennustetiedot
 - a. Kansallisen säämallin tuorein ennuste, joka sisältää muun muassa pintasää tietoja tunnin välein 2 vuorokauden ajalle
 - b. Merimallien (meriveden korkeus, virtaus ja aalto) tuoreimmat ennusteet
 - c. Ilmastonmuutosskenaariot 30-vuotisjaksoille 2010–2039, 2040–2069 ja 2070–2099 (keskimääräiset lämpötilan ja sateen muutosarvot).

Tietoaineistoa avataan vaiheittain. Ajantasainen tieto avatuista aineistoista löytyy aina Ilmatieteen laitoksen avoimen datan verkkosivuilta (<https://ilmatieteenlaitos.fi/avoin-data>). [11]

Avoimen datan käyttö vaatii käyttäjältä rekisteröitymisen verkkopalveluun osoitteessa <https://ilmatieteenlaitos.fi/avoin-data>. Rekisteröityessään käyttäjä hyväksyy Ilmatieteen laitoksen avoimen datan lisenssin, joka soveltaa Creative Commons – Nimeä 4.0 lisenssiä (CC BY 4.0). Lisenssi koskee avoimen datan verkkopalvelussa olevaa aineistoa. Rekisteröityessä käyttäjä saa käyttöönsä myös palvelun vaatiman tunnistevaimen (fmi-api-key). Tunnisteavain on muotoa 722ca3d1-1bb3-493b-a371-786abcdefghi ja sillä voi tehdä maksimissaan 20 000 kyselyä vuorokaudessa. Tunnisteavain pitää olla sisällytettynä palvelulle lähetettävässä kyselyssä.

Osa Ilmatieteen laitoksen palveluista on maksullisia. Tällaisia palveluja ovat esimerkiksi tietoaineistojen räätälöinti, palveluiden tuottaminen sekä muiden kuin avattavien tietoaineistojen käyttö. Mikäli asiakas haluaa, että Ilmatieteen laitos hakee ja toimittaa tietoaineistoa asiakkaan käyttöön, joutuu asiakas tällöin maksamaan toimituksesta aiheutuvat kustannukset. Yleinen sääpalvelu sekä myös varoituspalvelut tulevat aina olemaan maksuttomia. [11]

Tässä työssä käytettiin asemakohtaista ennustetietoja seuraavan 24 tunnin ajalle.

2.6.1 Saatavissa olevat sääennustetiedot

Ilmatieteen laitoksen avoimiin aineistoihin käytetään kansallista säämallin HIRLAM -sääennustetietoa. HIRLAM -säämallia ajetaan 4 kertaa vuorokaudessa kello 00.00, 06.00, 12.00 ja 18.00. Jokaisessa säämallissa ennustepituus on 54 tuntia.

Avoimen datan rajapinnasta on saatavilla aina uusimman säämalliajon tietoja. Saatavana ovat tärkeimmät pintasääsuureet, kuten esimerkiksi 2 metrin lämpötila, 2 metrin suhteellinen kosteus, ilmanpaine merenpinnan tasolla, 10 metrin tuulen nopeus, 10 metrin tuulen suunta, 10 metrin vaakasuuntaiset tuulikomponentit, kokonaispilvisuus ja sademäärä. Paine pintasuureista on saatavilla muun muassa lämpötila, suhteellinen kosteus, kastepistelämpötila, tuulen suunta, tuulen nopeus, tuulen vaakakomponentit ja geopotentiaalikorkeus. [11]

2.6.2 Tiedon hakeminen palvelusta

Avointa dataa voidaan hakea palvelusta osoitteesta <http://data.fmi.fi/wfs>. Palvelu on WFS 2.0 -palvelu ja sen toteuttaa WFS Simple Profile. Kyselyt tulee tehdä tallennettujen kyselyiden avulla. Palvelu vaatii rekisteröitymisen yhteydessä saadun tunnistevaimen käyttöä. Seuraavassa linkissä on yksi esimerkki valmiista kyselyistä, jolla voidaan tiedustella palveluun tallennettuja valmiita kyselyitä: [12]

<http://data.fmi.fi/fmi-apikey/<tuoteavaintähän>/wfs?request=listStoredQueries&>

Kaikkia tietoja voi hakea kahdessa eri formaatissa: Multipointcoverage- ja Timevaluepair-tyyppisenä. Multipointcoverage on tiiviimpi muoto, kun taas Timevaluepair on täysin INSPIRE-yhteensopiva. Näitä tukevat muun muassa uusimmat Excel-ohjelmat. [12]

Valmiita tallennettuja kyselyitä ja niissä olevia parametreja voi hakea Ilmatieteen laitoksen avoimen datan verkkosivuilta.

2.6.3 Palvelusta saatava vastaus

Ilmatieteen laitoksen avointa dataa jaetaan OGC -standardien mukaisessa muodossa. Tässä työssä käytetty kysely palauttaa tiedon OGC Catalog Service for Web (CSW) -muodossa. Käytännössä tämä tarkoittaa sitä, että työssä tehty ohjelma (Client) lähettää CSW-pyyntöön Ilmatieteen laitoksen palvelimelle, joka lähettää vastauksen pyyntöön XML-muodossa. [13]

```

▼<om:featureOfInterest>
  ▼<sams:SF_SpatialSamplingFeature gml:id="enn-s-1-1">
    ▼<sam:sampledFeature>
      ▼<target:LocationCollection gml:id="sampled-target-1-1">
        ▼<target:member>
          ▼<target:Location gml:id="forloc-geoid-658225-pos">
            <gml:identifier codeSpace="http://xml.fmi.fi/namespace/stationcode/geoid">658225</gml:identifier>
            <gml:name codeSpace="http://xml.fmi.fi/namespace/locationcode/name">Helsinki</gml:name>
            <gml:name codeSpace="http://xml.fmi.fi/namespace/locationcode/geoid">658225</gml:name>
            <target:representativePoint xlink:href="#point-658225"/>
            <target:country codeSpace="http://xml.fmi.fi/namespace/location/country">Finland</target:country>
            <target:timezone>Europe/Helsinki</target:timezone>
            <target:region codeSpace="http://xml.fmi.fi/namespace/location/region">Finland</target:region>
          </target:Location>
        </target:member>
      </target:LocationCollection>
    </sam:sampledFeature>
  </sams:shape>
  ▼<gml:MultiPoint gml:id="sf-1-1">
    ▼<gml:pointMembers>
      ▼<gml:Point gml:id="point-658225" srsName="http://www.opengis.net/def/crs/EPSSG/0/4258" srsDimension="2">
        <gml:name>Helsinki</gml:name>
        <gml:pos>60.16952 24.93545</gml:pos>
      </gml:Point>
    </gml:pointMembers>
  </gml:MultiPoint>
</sams:SF_SpatialSamplingFeature>
</om:FeatureOfInterest>

```

Kuva 7. Esimerkki Ilmatieteen laitoksen palvelimelta tulevasta XML-tiedoston muodosta.

2.7 Nord Pool

Tässä työssä tutkittiin, miten Spot-markkinaa voisi hyödyntää edullisimman sähkön hinnan hyödyntämiseen. Lopullisena tavoitteena oli hakea palvelusta ajat, jolloin sähkön hinta on kalleinta ja halvinta. Kalleimman tunnin aikana pitäisi välttää lämmitystä ja halvimmalla sähkön hinnalla aikana voisi käyttäjälle suositella esimerkiksi saunomista tai muuta vastaavaa paljon energiaa kuluttavaa tekemistä.

Nord Pool on Pohjoismaiden ja Baltian maiden yhdessä omistama osakeyhtiömuotoinen tyypillinen raaka-ainepörssi, sähköpörssi. Sillä on kolme keskeistä kauppapaikkaa: [14]

1. Spot-markkina: seuraavan päivän sähkön hinta tunnin tarkkuudella
2. Johdannaismarkkina: sähkön finanssitilanteiden vaihto

3. Elbas-tasesäätömarkkina

2.7.1 Spot-markkinat

Sähkön hinta Spot-markkinoilla määräytyy kysynnän ja tarjonnan mukaan. Sähkö on lähes aina eri hintaista eri päivinä ja tunteina. Keskeisiä toimijoita Spot-markkinoilla ovat sähkön tarjoajat ja sähkön ostajat. Sähkön tarjoajia ovat voimalaitoksia omistavat sähköntuottajat. Sähkön ostajia ovat sähkön vähittäismyyjät ja suuret teollisuuslaitokset, jotka ostavat sähkön suoraan pörssistä. [14]

2.7.2 Spot-hinnan määräytyminen

Spot-markkinoilla toimijat lähettävät ennen klo 13.00 (Suomen aikaa) Nord Poolille oman tarjouksensa sähkön osto- ja myyntihinnasta sekä ostettavan tai myytävän sähkön määrästä kullekin tunnille. Nord Pool laskee näiden tarjousten perusteella kysynnän ja tarjonnan lain perusteella sähkön spot-hinnat seuraavalle päivälle. [14]

2.7.3 Spot-hintatietohaun kustannukset

Nord Poolin tarjoaa erilaisia palveluja, joiden avulla käyttäjät voivat hakea energiamarkkinatietoja. Mikäli spot-hintatietoja halutaan hakea esimerkiksi FTP-palvelinta (<ftp.nordpoolspot.com>) hyväksikäyttäen, perii Nord Pool tästä maksun. Vuotuinen maksu päivänsisäisen hinnan tarjoavaan FTP-palvelimeen maksaa 1 600 € vuonna 2016. Nord Pool -asiakkaat, oppilaitokset ja järjestelmäkehittäjät saavat tästä hinnasta 50 % alennuksen. Opiskelijoille ja Nord Pool jäsenille palvelu on ilmainen. Hintojen hakemiseksi tarvitaan FTP-palvelimelle käyttäjätunnus ja salasana. Nord Pool toimittaa ne, kun käyttäjä on joko maksanut heille vuosimaksun tai todistanut hyväksytysti olevansa opiskelija.

FTP-palvelussa sähkön hinnat ovat tarjolla kahdessa erilaisessa muodossa. Kuvassa 8 on esimerkki xls-tiedostosta ja kuvassa 9 esimerkki sdv-tiedostosta.

Helsinki-prices in Elspot pr hour for 2016
 Hourweighted price pr. hour (EUR/MWh)
 Source: Nord Pool

| Date | Hour | 1 | 2 | 3A | 3B | 4 | 5 | 6 | 7 |
|-----------|-------|-------|-------|----|----|-------|-------|-------|-------|
| 1.1.2016 | 16.39 | 16,04 | 15,74 | | | 15,57 | 15,47 | 15,75 | 15,36 |
| 2.1.2016 | 16,47 | 15,89 | 15,03 | | | 13,19 | 12,69 | 12,94 | 13,59 |
| 3.1.2016 | 17,95 | 15,87 | 15,76 | | | 15,59 | 15,69 | 15,83 | 14,98 |
| 4.1.2016 | 15,96 | 15,81 | 15,56 | | | 15,57 | 15,92 | 27,16 | 34,03 |
| 5.1.2016 | 17,68 | 17,33 | 17,24 | | | 17,32 | 17,66 | 32,99 | 34,08 |
| 6.1.2016 | 27,97 | 23,61 | 21,77 | | | 22,42 | 25,99 | 32,94 | 34,02 |
| 7.1.2016 | 29,10 | 27,16 | 27,14 | | | 27,92 | 32,66 | 40,76 | 57,02 |
| 8.1.2016 | 30,12 | 28,56 | 27,53 | | | 27,94 | 31,35 | 49,12 | 58,06 |
| 9.1.2016 | 28,01 | 25,64 | 23,85 | | | 25,70 | 26,62 | 27,69 | 27,37 |
| 10.1.2016 | 21,32 | 19,93 | 19,65 | | | 19,43 | 19,25 | 20,43 | 23,37 |
| 11.1.2016 | 21,37 | 20,04 | 19,78 | | | 19,36 | 30,02 | 39,92 | 57,59 |
| 12.1.2016 | 21,11 | 19,81 | 19,10 | | | 18,41 | 25,90 | 30,96 | 34,06 |
| 13.1.2016 | 21,00 | 19,66 | 18,72 | | | 18,66 | 24,69 | 33,65 | 34,95 |
| 14.1.2016 | 23,51 | 22,58 | 22,22 | | | 22,03 | 25,71 | 33,32 | 34,53 |
| 15.1.2016 | 24,58 | 24,04 | 23,28 | | | 24,00 | 33,13 | 42,40 | 46,66 |
| 16.1.2016 | 26,49 | 25,70 | 24,84 | | | 24,34 | 24,03 | 24,71 | 24,18 |
| 17.1.2016 | 21,88 | 20,25 | 20,76 | | | 20,97 | 20,89 | 20,94 | 20,12 |

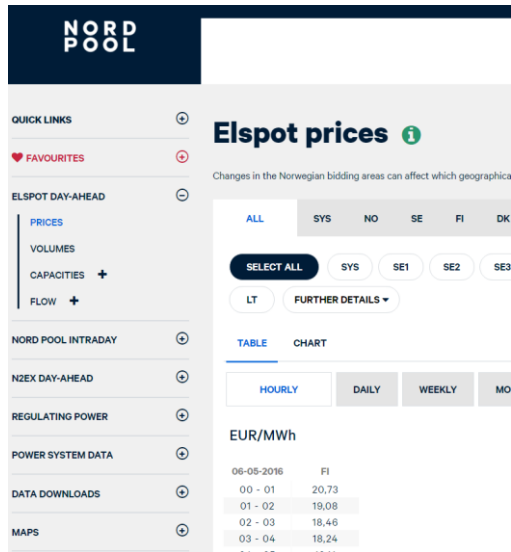
Kuva 8. Nord Poolin Elspot-hinnat xls-muodossa.

```
Helsinki-prices in Elspot pr hour for 2016;;
Hourweighted price pr. hour (EUR/MWh);;
Source: Nord Pool;;
;;
Date;Hour;
;1;2;3A;3B;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21;22;23;24;Kjoredag;
01.01.16;16,39;16,04;15,74;;15,57;15,47;15,75;15,36;15,37;15,19;15,39;15,78;16,
02.01.16;16,47;15,89;15,03;;13,19;12,69;12,94;13,59;17,37;19,34;19,96;19,91;16,
03.01.16;17,95;15,87;15,76;;15,59;15,69;15,83;14,98;15,53;15,79;16,36;17,15;17,
04.01.16;15,96;15,81;15,56;;15,57;15,92;27,16;34,03;34,47;34,46;34,05;34,08;34,
05.01.16;17,68;17,33;17,24;;17,32;17,66;32,99;34,08;34,44;34,41;34,07;34,23;34,
06.01.16;27,97;23,61;21,77;;22,42;25,99;32,94;34,02;34,05;34,06;34,09;34,97;37,
07.01.16;29,10;27,16;27,14;;27,92;32,66;40,76;57,02;72,26;99,93;79,60;74,91;61,
08.01.16;30,12;28,56;27,53;;27,94;31,35;49,12;58,06;100,01;99,94;76,04;69,09;50,
09.01.16;28,01;25,64;23,85;;25,70;26,62;27,69;27,37;29,36;28,99;30,45;29,95;30,
10.01.16;21,32;19,93;19,65;;19,43;19,25;20,43;23,37;27,00;29,13;32,93;32,93;32,
11.01.16;21,37;20,04;19,78;;19,36;30,02;39,92;57,59;67,85;60,04;55,63;41,94;40,
12.01.16;21,11;19,81;19,10;;18,41;25,90;30,96;34,06;35,39;39,53;41,00;38,66;36,
13.01.16;21,00;19,66;18,72;;18,66;24,69;33,65;34,95;38,24;36,19;35,26;31,45;33,
14.01.16;23,51;22,58;22,22;;22,03;25,71;33,32;34,53;69,91;88,61;69,98;49,60;50,
15.01.16;24,58;24,04;23,28;;24,00;33,13;42,40;46,66;119,92;134,43;150,01;58,00;
16.01.16;26,49;25,70;24,84;;24,34;24,03;24,71;24,18;26,39;26,58;31,63;33,39;33,
17.01.16;21,88;20,25;20,76;;20,97;20,89;20,94;20,12;22,35;23,52;24,58;26,44;27,
18.01.16;23,96;23,47;23,56;;25,74;34,69;50,02;66,15;95,06;94,39;95,02;51,41;50,
```

Kuva 9. Nord Poolin Elspot-hinnat sdv-muodossa.

Sähkön hinnat on esitetty megawattitunteina ja ne ovat tarjolla 4 eri valuutassa: Tanskan kruunuina (DKK), Euroina (EUR), Ruotsin kruunuina (SEK) ja Norjan kruunuina (NOK).

Samat sähkön hinnat löytyvät myös Nord Poolin verkkosivuilta osoitteesta www.nordpoolgroup.com. Esimerkki verkkosivuilta on esitetty kuvassa 10.



Kuva 10. Esimerkki hinnoista Nord Poolin verkkosivuilla.

3 OHJELMAN SUUNNITTELU JA OHJELMOINTI

3.1 Ongelmat, joita lähdettiin ratkaisemaan

Monissa nykyaikaisissa automaatiojärjestelmissä lämpötilan säätö ei ehdi reagoida nopeisiin lämpötilanmuutoksiin, mikä yleensä johtaa siihen, että asunnon lämpötila pääsee putoamaan liian alas tai nousemaan liian ylös. Lämpötilan säätö tehdään nykyisissä järjestelmissä vallitsevan ulko- ja/tai sisälämpötilan mukaan käyttäen esimerkiksi talon ulkopuolella olevaa lämpötila-anturia yhdessä sisällä olevan huonetermostaatin kanssa. Lämpötilan nostaminen tai laskeminen esimerkiksi lattialämmitystaloissa on hitaampaa suuren lämmitysmassan vuoksi ja siksi lämpötilan säätö ei ehdi reagoida nopeaan lämpötilanmuutokseen. Tästä johtuen taloa lämmitetään usein liian pitkään ja samalla kallista energiaa hukataan yllälämmittämisen seurauksena. Nykyaikaisilla ilmastointikoneilla voidaan lämmittää tai viilentää asuntoa nopeastikin. Turhaa lämmitystä ja viilennystä tulee kuitenkin välttää, eli lämmitystä ja viilennystä tehdään vain silloin, kun se oikeasti on tarpeellista.

Järjestelmään haluttiin saada tieto tulevan säätilan muutoksista 36 tuntia ennen säätilan muuttumista, jotta järjestelmä pystyisi ennakoimaan tulevat muutokset ajoissa ja näiden tietojen mukaisesti optimoimaan talossa tarvittavan lämmityksen ja viilennyksen. Tietoa tulevista säätilan muutoksista voidaan käyttää muun järjestelmän nopeasti tapahtuvaan säätämiseen ja näin varautua esimerkiksi seuraavan päivän lämpimään ilmaan niin, että normaalia lämmitystä ei yön aikana tarvitse käyttää. Sen sijaan yöaikana voidaan ilmastointikoneella varastoida viileää ilmaa sisään. Öinen turha lämmittäminen sekä seuraavan päivän ilmastointikoneella tapahtuva viilentäminen minimoidaan. Näin etukäteen saadulla sääennusteella voidaan minimoida sähkönkulutus. Tätä ei voida tehdä, jos sää-tödata perustuu vain vallitsevaan säätilaan.

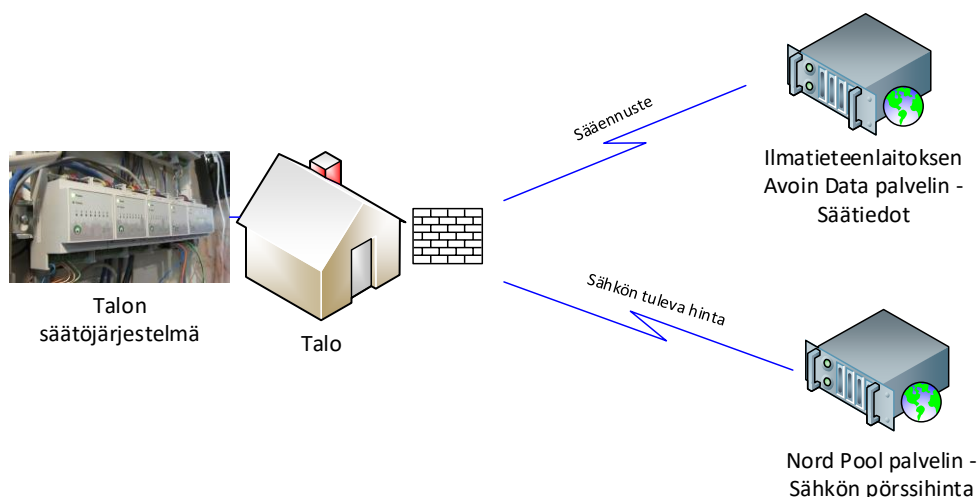
Jotta järjestelmä olisi mahdollisimman pitkälle optimoitu, haluttiin ottaa käyttöön myös alhainen energian hinta. Tavoitteena oli käyttää sähköä silloin, kun se on halvimmillaan. Tätä varten haluttiin saada reaaliaikainen tieto seuraavan päivän energian hinnasta, jotta lämmitys tai viilennys voidaan tehdä edullisen sähkön hinnan aikana.

Yrityksen tavoitteena oli tehdä järjestelmä, joka minimoisi energian kulutuksen automaattisesti tinkimättä mukavuudesta ja antaisi käyttäjälle hyvän käyttäjäkokemuksen.

Tässä työssä keskitytään kahteen pääongelmaan. Ensimmäisenä selvitetään, miten lämpötilatieto saadaan haettua automaattisesti tulevalle 36 tunnille ja millaisen ohjelman logiikka tarvitsee. Toisena selvitetään se, miten saadaan haettua reaaliaikainen sähkön hinta tulevalle vuorokaudelle. Sähkön hintaa varten selvitetään tapa hakea tuleva sähkön hinta. Itse ohjelmaa ei tässä työssä toteuteta, vaan sen tekeminen jätetään myöhempään ajankohtaan. Ohjelman tarvitsemat komponentit mietitään valmiiksi tulevaisuuden ohjelmointia varten.

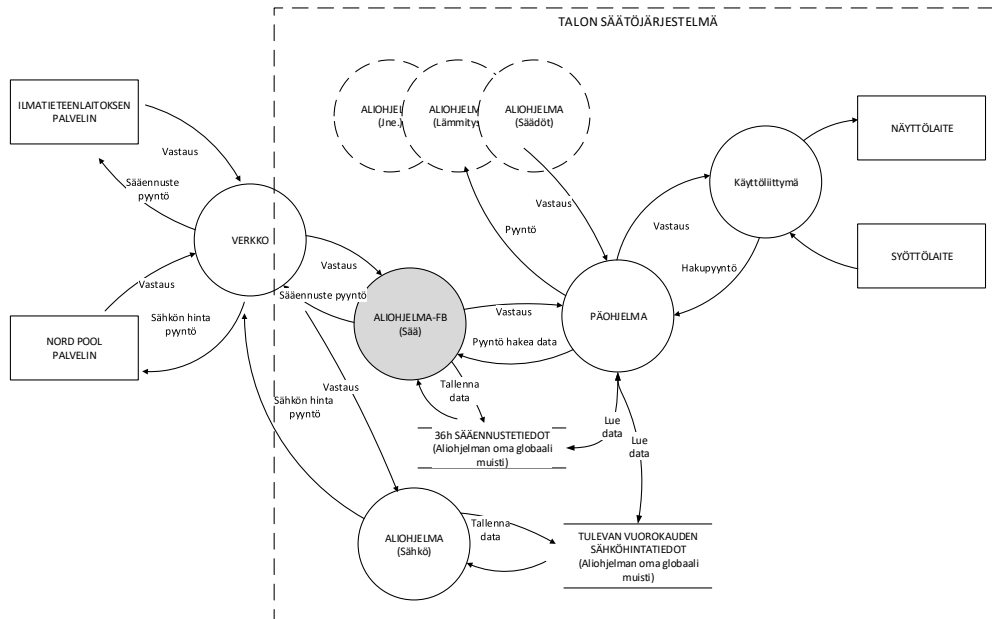
3.2 Arkkitehtuuri

Korkealla tasolla arkkitehtuuri jakautuu kuvan 11 mukaan 3 eri pääosaan: Kodin säätöjärjestelmään, Ilmatieteen laitoksen Avoin data -palveluun sekä Nord Poolin tarjoamaan Elspot -palveluun. Talon säätöjärjestelmä kysyy automaattisesti tulevan säätilan sekä sähkön hinnan palvelimilta käyttäen verkkoyhteyttä. Palvelimet palauttavat vastausviestissä 36 tunnin sääennusteen ja seuraavan päivän sähkön tuntihinnat talon säätöjärjestelmälle. Sääntöjärjestelmä käsittelee ja tallentaa tiedon. Talon säätöjärjestelmä tekee tarvittavat säädöt näiden tietojen ja muiden saatavilla olevien tietojen pohjalta.



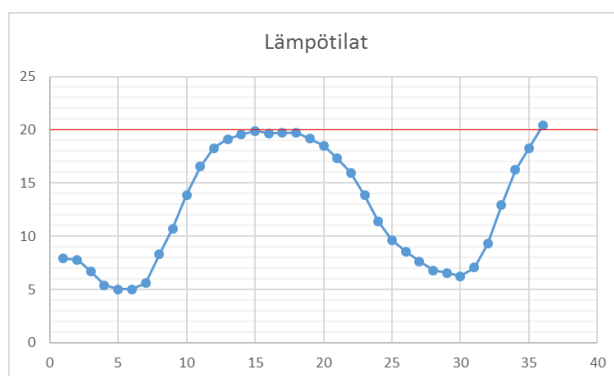
Kuva 11. Korkean tason arkkitehtuuri.

Talon säätöjärjestelmän ohjelma jakautuu useampaan eri lohkokon kuvan 12 mukaisella tavalla: pääohjelmaan ja useampaan aliohjelmaan. Kuvassa 12 aliohjelma sääennustedan hakemiseen on merkitty harmaalla värillä.



Kuva 12. Vuokaavio ohjelman toiminnasta.

Aliohjelma käsittelee saadun vastauksen, tallentaa tuloksen muistiin ja välittää halutun tiedon pääohjelmalle. Sää-aliohjelmalta välitetään pääohjelmalle tieto siitä, ylittääkö seuraavan päivän lämpötila pääohjelman asettaman lämpötilan hakurajan. Jos tämä lämpötilaraja ylitetään, niin aliohjelma asettaa Raja_Ylitetty-bitin, jonka perusteella pääohjelma voi aloittaa yöllä esimerkiksi jäähdytyksen tai muun toimenpiteen. Kuvassa 13 pyydetty lämpötilaraja on 20 °C ja haettavasta datasta näkyy miten se ylittää asetetun rajan. Pääohjelma tekee lämpötilan hakemisen kello 21.00, jotta sääennuste olisi mahdollisimman tarkka ja tarvittavat säätötoimenpiteet ehditään tehdä vielä saman yön aikana.



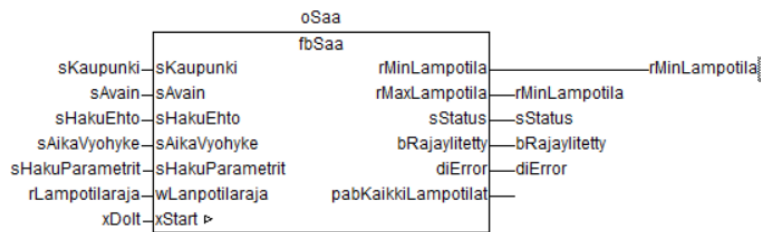
Kuva 13. Ilmatieteen laitokselta haettu lämpötilaennuste.

3.3 Sää-aliohjelma

Aliohjelma tehdään järjestelmään Function Block diagrammina, jota pääohjelma kutsuu, kun säätilatietoa pitää hakea. Pääohjelma välittää aliohjelmalle parametrit, joiden mukaan hakua lähdetään tekemään.

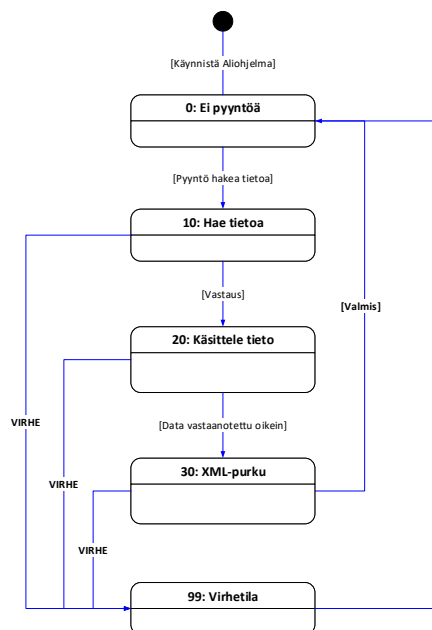
3.3.1 Aliohjelman parametrit

Aliohjelman sää Function Block muodostuu input- ja output-arvoista sekä ohjelmallisesta koodista, joka suorittaa itse tehtävän [Kuva 14]



Kuva 14. Aliohjelman sisäänmenevät ja ulostulevat parametrit.

3.3.2 Tilakone



Kuva 15. Sää-aliohjelman tilakone.

Sää-aliohjelman sisälle on ohjelmoitu tilakone [Kuva 15]. Tilakone tekee tarvittavat viestit, lähettää ne, käsittelee virhetilanteet ja purkaa vastaanotetut viestit [Kuva 16].

```

CASE wTilakone OF
0: (*** INITIALISOI *****)
  IF xStart THEN
    (* Rakenna haettava Uri-osoite *)
    sUri := 'fmi-apikey';
    (*lisää apikey*)
    sUriHelp := HTTP_URIEncode(sAvain);
    sUri := CONCAT(sUri, sUriHelp);
    sUri := CONCAT(sUri, 'wfs?request=getFeature&storedquery_id=');

    (* lisää valmis hakuehto - sHakuEhto *)
    sUriHelp := HTTP_URIEncode(sHakuEhto);
    sUri := CONCAT(sUri, sUriHelp);

    (*lisää timezone - sAikaVyohyke*)
    sUri := CONCAT(sUri, '&timezone=');

    sUriHelp := HTTP_URIEncode(sAikaVyohyke);
    sUri := CONCAT(sUri, sUriHelp);

    (*lisää city - sCityState *)
    sUri := CONCAT(sUri, '&place=');

    sUriHelp := HTTP_URIEncode(sKaupunki);
    sUri := CONCAT(sUri, sUriHelp);

    (*lisää parametrit - sHakuParametrit*)
    sUri := CONCAT(sUri, '&parameters=');

    sUriHelp := HTTP_URIEncode(sHakuParametrit);
    sUri := CONCAT(sUri, sUriHelp);

    (* Tee HTTP_GET viesti *)
    sHttp := httpData := ΔNP(sUri);
  
```

Kuva 16. Koodilistaus, tilakoneen alku.

3.3.3 Haun tekeminen

Pääohjelman pitää välittää aliohjelmalle seuraavat tiedot, jotta haku palvelusta onnistuisi oikein:

- Aikavyöhyke (tämän perusteella määritellään kellonajat)
- Kaupunki tai kunta, johon sääennuste halutaan hakea
- Haettavat tiedot (esimerkiksi: Lämpötila (*Temperature*))

Seuraavassa on esimerkki hausta, joka hakee 36 tunnin sääennusteen Tarvasjoelle 1 tunnin välein aikavyöhykkeellä Europe/Helsinki:

http://data.fmi.fi/fmi-apikey/AVAIN/wfs?request=getFeature&storedquery_id=fmi::forecast::hirlam::surface::point::timevaluepair&timezone=Europe/Helsinki&place=Tarvasjoki¶meters=temperature

Pääohjelma käynnistää haun asettamalla xDolt-muuttujan arvoksi TOSI, jolloin aliohjelman haku käynnistyy.

3.3.4 Vastauksen purkaminen

Sääennusteviesti Ilmatieteen laitoksen rajapinnalta on XML-muotoista tietoa.

Vastausviesti tarkastetaan ja puretaan tilakoneessa. Viesti hylätään, mikäli siinä on virheitä. Tällaisia virheitä voivat olla esimerkiksi vääränlainen tieto viestissä tai viestiä ei ole kokonaisuudessaan vastaanotettu.

XML-tiedon purkamiseksi on tilakoneeseen rakennettu oma käsittelijä, joka purkaa pyydetyn tiedon XML-viestistä ja tallentaa sen muistialueelle, josta pääohjelma tai käyttöjärjestelmä voi sen lukea. Muistialue on sellaisessa paikassa, että muistin sisältö säilyy muuttumattomana esimerkiksi sähkökatkotilanteissa. Tällä tavalla pienet sähkökatkokset eivät sekoita järjestelmän toimintaa, vaan laite toimii aina viimeisimpien tietojen pohjalta. Aliohjelma ilmoittaa pääohjelmalle, kun viesti on täysin vastaanotettu, purettu ja tallennettu. Myöskin vikatilanteista ilmoitetaan pääohjelmalle.

3.4 Sähkön hinnan hakeminen

Sähkön hinnan hakemiseksi selvitettiin haettavan tiedoston tiedostomuoto ja tarvittavat output-parametrit aliohjelmasta pääohjelmalle.

Tiedostomuodossa päädyttiin siihen, että tulevaisuudessa tehtävässä ohjelmassa tul- laan käyttämään sdv-formaattia, sillä sen käsittely logiikkapiirissä nähtiin yksinkertaisem- maksi.

Aliohjelman pitää toteuttaa hieman vastaavanlainen tilakone, jota on käytetty jo sääen- nusteen käsittelyssä. Tilakoneessa pitää etsiä seuraavan päivän sähkönhinnoista kallein ja halvin aika. Näillä parametreilla välitetään pääohjelmalle tieto siitä, mihin aikaan kan- nattaa tai ei kannata kuluttaa sähköä. Tämä tieto näytetään myös loppukäyttäjälle. Out- put-parametrit voidaan nimetä esimerkiksi iKalleinAika ja iHalvinAika. Molemmat aikapa- rametrit ovat Int-tyyppisiä parametreja.

3.5 Tietoliikenne

3.5.1 Käytetyt verkkoprotokollat

Sääennusteen hakemiseksi käytetään Ilmatieteen laitoksen avoimen datan rajapintaa. Tästä rajapinnasta data haetaan HTTP_GET-viestillä. HTTP_GET-viesti käyttää järjestelmässä TCP- ja System Socket -rajapintoja.

Nord Pool -palvelimen rajapinta toimii FTP -protokollaa käyttäen.

3.5.2 Tietoturva

Sääennustedata haetaan julkisesta ilmaisesta palvelusta. Haettava data on kaikille avointa ja tästä syystä tietoverkossa välitettävässä datassa ei ole mitään tietoturvan kannalta kriittistä tietoa. Tätä dataa ei salata millään tavalla.

Talon säätöjärjestelmä on kytketty ulkoiseen tietoverkkoon logiikkakortilla. Siksi on erittäin tärkeää huolehtia siitä, että kukaan ulkopuolinen ei pääse käsiksi tähän säätöjärjestelmään. Kyseisessä järjestelmässä yhteys on suojattu nykyaikaisella palomuurilla, järjestelmä on suojattu salasanoilla.

3.6 Testaus

Testauksessa testattiin yhteyden toimivuus talon säätölaitteesta Ilmatieteen laitoksen avoimen datan palveluun sekä aliohjelman toimivuus. Testaukset suoritettiin yrityksen tiloissa käyttäen oikeaa lopullista laiteympäristöä.

Aliohjelman testausta varten kirjoitettiin erilaisia yksikkötestejä, jotta voitiin varmistua aliohjelman toimivuudesta. Yksikkötestejä tehtiin erilaisten hakuehtojen testaamiselle, XML viestin purkamiselle sekä erilaisille vikatilanteille.

Yhteys testattiin ajamalla ohjelmaa niin, että HTTP_Get-viesti lähetettiin ohjelmalta ulos Ilmatieteen laitoksen avoimen datan palvelimelle ja odotettiin vastausviestiä palvelimelta. Hakuviesti testattiin ensin tietokoneen selaimella, jotta varmistuttiin oikeanlaisesta viestirakenteesta. Tämän jälkeen sama viesti lähetettiin aliohjelmalta. Vastauksen tuli palautua takaisin ilman virheitä.

Alussa ongelmia tuotti väärä IP-osoite ohjelmoitavassa logiikassa. Yhteys alkoi toimia normaalisti, kun IP-osoite oli korjattu.

Tämän jälkeen testattiin aliohjelmaa. Liitteessä 1 on esimerkki testistä, jossa ajetaan haku. Palautuneesta XML-viestistä tilakone etsii korostetut tekstit ja asettaa bRajaylitetty-lipun päälle, koska tässä käytetty 19 °C lämpötila ylittyy seuraavana päivänä.

Virhetilanteita testattiin muun muassa katkaisemalla yhteys palvelimelle, antamalla aliohjelmalle väärää Input-parametreja sekä tekemällä vääränlaisia hakuviestejä, jotka palauttivat virheilmoituksia avoimen datan palvelusta.

Kaikkien näiden testien jälkeen voitiin todeta, että ohjelmisto toimii juuri sille asetettujen tavoitteiden mukaisesti.

Tässä esimerkkitesti. Kyselyyn saatu vastaus löytyy Liitteestä 1.

Paikkakunta: Tarvasjoki

Lämpötilaraja: 19 astetta

Hakuviesti (HTTP_GET): http://data.fmi.fi/fmi-apikey/AVAIN/wfs?request=getFeature&storedquery_id=fmi::forecast::hirlam::surface::point::timevaluepair&timezone=Europe/Helsinki&place=Tarvasjoki¶meters=temperature

TULOS: Haku onnistui ja bRajaylitetty-bitti on päällä – **Testi OK**

4 PÄÄTELMÄT

Tässä työssä oli tarkoituksena suunnitella sekä kehittää talon säätöjärjestelmän ohjelmoitavaan logiikkaan aliohjelma, joka ulkoista palvelua käyttämällä hakee säätiedot tulevalle 36 tunnille. Samalla piti selvittää ja alustavasti suunnitella, millä ulkoisella palvelulla logiikalle saadaan tulevan vuorokauden sähkönhintatiedot kokonaissäätöjärjestelmän käyttöön.

Työ jaettiin eri loogisiin osioihin, joita lähdettiin selvittämään. Osioita olivat talon säätöjärjestelmään tehtävä aliohjelma, ulkoinen palvelu sääennustetiedon hakemiseksi ja ulkoinen palvelu tulevan sähkönhintatiedon hakemiseksi.

Itse ohjelman tekemiseen käytettiin CoDeSys-ohjelmointiympäristöä ja WAGON 775-881-logiikkakontrolleria, sillä näitä käytettiin jo Timil Oy:ssä. Sääennustetietojen saamiseksi työssä käytettiin Ilmatieteen laitoksen avoimen datan rajapintaa, joka on ilmainen ja joka tarjoaa kattavaa säädataa. Sähkön hintatietojen hakemisesta tiedettiin ennalta, että Nord Pool tarjoaa sähkön pörssihintatietoja seuraavalle päivälle. Tästä syystä Nord Pool valikoitui tavoitepalveluksi sähkön hintatietojen saamiseksi. Tavoitteena ei ollut toteuttaa aliohjelmaa sähkön hintatietojen hakemiseksi, vaan tutkia miten ja millä kustannuksilla sitä voidaan jatkossa toteuttaa.

Itse työ oli erittäin mielenkiintoista ja haastavaa, ja sen toteuttamiseen tarvittava työmäärä tuntui alussa sopivalta. Työkalut ja palvelut olivat kuitenkin täysin tuntemattomia, joten asian opetteluun ja työn tekemiseen meni huomattavasti enemmän aikaa kuin alun perin oli suunniteltu.

Työn aikana huomattiin, että esimerkiksi CODESYS-simulaatioympäristöstä ei saatu suoraa internetyhteyttä Ilmatieteen laitoksen avoimen datan testaukseen. Tätä varten olisi pitänyt olla käytössä oikea logiikkakontrolleri, joka oli kallis vain tähän käyttöön hankittavaksi. Ohjelman kehittämistä ja testausta varten työssä piti lainata Timil Oy:n omassa käytössä olevaa logiikkakontrolleria, joka teki tekemisestä monimutkaisempaa ja aikaa vievää.

Ilmatieteen laitoksen avoin data on kattava, mutta monessa suhteessa saatava tieto on monimutkaista ja helpollakin kyselyllä tietoa tulee paljon. Usein XML-tiedosto sisältää paljon dataa ja oikean sääennustehakuehdon löytäminen olikin monien kokeilujen takana. Valmista XML-parseria ei WAGO -kontrolleriin löytynyt, joten se piti tehdä itse.

Yhdessä monimutkaisen XML-tiedoston kanssa tämä oli yksi eniten aikaa kuluttaneista tehtävistä.

Nord Pool -palvelusta saatava tieto osoittautui erittäin käyttökelpoiseksi. Huonona puoleena on se, että kaupallisessa käytössä tuon datan saaminen FTP-palvelusta on pienelle yrittäjälle kallista. Tämän työn tuloksena syntynyttä Ilmatieteen laitoksen sää-aliohjelmalla jatkossa käyttämään sähkönhintatietojen hakemiseen tarvittavan ohjelman tekemisessä.

Työn aikana saatiin tehtyä kaikki ne loogiset osiot, jotka alussa tavoitteiksi asetettiin. Työn aikana nousi esiin asioita, joita voi jatkossa kehittää. Tällaisia asioita olivat muun muassa aliohjelma sähkönhintatietojen hakua varten, käyttöliittymä sähkönhintatiedoille sekä sääennustetiedoille. Palvelun käyttöön tarvittava paikkatieto voisi tulla myös suoraan GPS:ltä, jolloin paikkatiedon saaminen järjestelmälle olisi automaattista. Koko järjestelmän valvontaan ja säätämiseen voisi tehdä erillisen mobiilisovelluksen tai pilvipalvelun.

LÄHTEET

- [1] Sillvan Niko. 2016. Verkkosovellus Älykkääseen Laiteohjeukseen. [opinnäytetyö]. Saatavissa: http://www.theseus.fi/bitstream/handle/10024/110350/Sillvan_Niko.pdf?sequence=1. (luettu: 19.10.2016)
- [2] Mäkelä Petri. 2014. Avoimen tuulitiedon hyödyntäminen tuulivoimaprojektin suunnitteluvaiheessa. [opinnäytetyö]. Saatavissa: https://theseus.fi/bitstream/handle/10024/76065/makela_petri.pdf?sequence=1. (luettu 19.10.2016)
- [3] Pelttari Sirpa. 2014. Avoin Data. [opinnäytetyö]. Saatavissa: http://www.theseus.fi/bitstream/handle/10024/86046/Pelttari_Sirpa.pdf;jsessionid=CBB05F3BE9CFC17234A8E02FCF6920BE?sequence=1. (luettu 19.10.2016)
- [4] PLCOpen, Introduction into IEC 61131-3 Programming Languages, [www-sivu]. Saatavissa: http://www.plcopen.org/pages/tc1_standards/iec_61131_3/. (luettu 12.10.2015)
- [5] Ohjelmitava logiikka, [www-sivu]. Saatavissa: https://fi.wikipedia.org/wiki/Ohjelmitava_logiikka (luettu 12.10.2015)
- [6] SKS Group, [www-sivu]. Saatavissa: <http://www.sks.fi/www/ohjelmistot&id=CODESYS>. (luettu 11.4.2016)
- [7] WAGO Suomi, [www-sivu]. Saatavissa: <http://www.wago.fi/wago-suomessa/lisaae-meis-tae/index.jsp>. (luettu 5.3.2016)
- [8] WAGO I/O Systems 750, Ethernet Programmable Fieldbus 750-881, [pdf-tiedosto]. Saatavissa: http://global.wago.com/media/2_products/q07500881_00000000_0en.pdf. (luettu 4.7.2015)
- [9] Valtiovarainministeriö, Avoin tieto, [www-sivu]. Saatavissa: <http://vm.fi/avoin-tieto>. (luettu: 10.4.2016)
- [10] Valtiovarainministeriö, Avoimen tiedon ohjelman 2013-2015 loppuraportti, [www-sivu]. Saatavissa: <http://vm.fi/julkaisu?pubid=6902>. (luettu 10.4.2016)
- [11] Ilmatieteen laitoksen avoin data, [www-sivu]. Saatavissa: <https://ilmatieteenlaitos.fi/avoin-data>. (luettu 5.3.2015)
- [12] Ilmatieteen laitoksen avoin data, Latauspalvelun pikaohje, [www-sivu]. Saatavissa: <http://ilmatieteenlaitos.fi/latauspalvelun-pikaohje>. (luettu 5.3.2015)
- [13] Catalog Service for the Web, [www-sivu]. Saatavissa: https://en.wikipedia.org/wiki/Catalog_Service_for_the_Web. (luettu 10.4.2016)
- [14] Nordpool, [www-sivu]. Saatavissa: https://fi.wikipedia.org/wiki/Nord_Pool. (luettu 5.3.2016)
- [15] Codesys, [www-sivu]. Saatavissa: <https://www.codesys.com/>. (luettu 5.3.2016)

Ilmatieteen laitoksen palvelimelta tullut XML-tiedosto

```

<wfs:FeatureCollection xmlns:wfs="http://www.open-
gis.net/wfs/2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-in-
stance" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:om="http://www.open-
gis.net/om/2.0" xmlns:omso="http://inspire.ec.europa.eu/schemas/omso/3.0" xmlns:ompr="http://in-
spire.ec.europa.eu/schemas/ompr/3.0" xmlns:gml="http://www.open-
gis.net/gml/3.2" xmlns:gmd="http://www.isotc211.org/2005/gmd" xmlns:gco="http://www.isotc211.org/2005/
gco" xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:gmlcov="http://www.opengis.net/gml-
cov/1.0" xmlns:sam="http://www.opengis.net/sampling/2.0" xmlns:sams="http://www.opengis.net/samplingS-
patial/2.0" xmlns:wml2="http://www.opengis.net/waterml/2.0" xmlns:target="http://xml.fmi.fi/names-
pace/om/atmosphericfeatures/1.0" timeStamp="2016-05-21T21:08:04Z" numberMatched="1" numberRetur-
ned="1" xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/2.0/wfs.xsd
http://www.opengis.net/gmlcov/1.0 http://schemas.opengis.net/gmlcov/1.0/gmlcovAll.xsd
http://www.opengis.net/sampling/2.0 http://schemas.opengis.net/sampling/2.0/samplingFeature.xsd
http://www.opengis.net/samplingSpatial/2.0 http://schemas.opengis.net/samplingSpatial/2.0/spatial-
SamplingFeature.xsd http://www.opengis.net/swe/2.0 http://schemas.opengis.net/sweCommon/2.0/swe.xsd
http://inspire.ec.europa.eu/schemas/omso/3.0 http://inspire.ec.europa.eu/schemas/omso/3.0/Specia-
lisedObservations.xsd http://inspire.ec.europa.eu/schemas/ompr/3.0 http://inspire.ec.europa.eu/sche-
mas/ompr/3.0/Processes.xsd http://www.opengis.net/waterml/2.0 http://schemas.opengis.net/wa-
term1/2.0/waterml2.xsd http://xml.fmi.fi/namespace/om/atmosphericfeatures/1.0
http://xml.fmi.fi/schema/om/atmosphericfeatures/1.0/atmosphericfeatures.xsd">
<wfs:member>
<omso:PointTimeSeriesObservation gml:id="WFS-tA0KHCah75zgvWEufbN0ugb.rI2JTowtIWbbpdOs2_1lx4efR060ae-
WzDtdOufXlmw48rplw36d3R066aduXth2dcvDDp5QtLFlz6d1TTty2oP4UrV0pqiMbHy51qRa-
FO06dNGTvwzsu7JU07ctqd.FK3dI4ITgzad2Sdlw8svPpWw7NOSQzM.Xfpyc6zGzNo4asXRry.e_1kv7.2Xl35aem-
HfsyxNLZl7ZdlTzwy1MzZvx4dmWtTNpvxpK7r0zOGttw.Imnn0w7seWlauhAa23Tz56d2ep18dKxp2Gcytu_Jl2VmaN-
PLZ2srhh5Ydta3TLt4ZeWHP15Zw5z6b.WXJx65eXm_pyV0hZtul06zb.WXhH59HTrRp5bM010659eWbDjyun-
XDfp3dHTrrp25e2HZ1y8Monk0Omnblp9MvCfkJodNo3La37stb0Xry38Mq.Rl2c907XpA---temperature">
<om:phenomenonTime>
<gml:TimePeriod gml:id="time-interval-1-1">
<gml:beginPosition>2016-05-21T22:00:00Z</gml:beginPosition>
<gml:endPosition>2016-05-23T09:00:00Z</gml:endPosition>
</gml:TimePeriod>
</om:phenomenonTime>
<om:resultTime>
<gml:TimeInstant gml:id="time-1-1">
<gml:timePosition>2016-05-21T12:00:00Z</gml:timePosition>
</gml:TimeInstant>
</om:resultTime>
<om:procedure xlink:href="http://xml.fmi.fi/inspire/process/hirlam"/>
<om:parameter>
<om:NamedValue>
<om:name xlink:href="http://inspire.ec.europa.eu/codeList/ProcessParameterValue/value/numeri-
calModel/analysisTime"/>
<om:value>
<gml:TimeInstant gml:id="analysis-time-1-1-temperature--">
<gml:timePosition>2016-05-21T12:00:00Z</gml:timePosition>
</gml:TimeInstant>
</om:value>
</om:NamedValue>
</om:parameter>
<om:observedProperty xlink:href="http://data.fmi.fi/fmi-apikey/A/meta?observableProperty=fore-
cast&param=temperature&language=eng"/>
<om:featureOfInterest>
<sams:SF SpatialSamplingFeature gml:id="enn-s-1-1-temperature">
<sam:sampledFeature>
<target:LocationCollection gml:id="sampled-target-1-1-temperature">
<target:member>
<target:Location gml:id="forloc-geoid-634851-pos-temperature">
<gml:identifier codeSpace="http://xml.fmi.fi/namespace/stationcode/geoid">634851</gml:identifier>
<gml:name codeSpace="http://xml.fmi.fi/namespace/locationcode/name">Tarvasjoki</gml:name>
<gml:name codeSpace="http://xml.fmi.fi/namespace/locationcode/geoid">634851</gml:name>
<target:representativePoint xlink:href="#point-634851-temperature"/>
<target:country codeSpace="http://xml.fmi.fi/namespace/location/country">Finland</target:country>
<target:timezone>Europe/Helsinki</target:timezone>
<target:region codeSpace="http://xml.fmi.fi/namespace/location/region">Lieto</target:region>
</target:Location>
</target:member>
</target:LocationCollection>
</sam:sampledFeature>
<sams:shape>
<gml:MultiPoint gml:id="foi-multipoint-1-1-temperature">
<gml:pointMembers>
<gml:Point gml:id="point-634851-temperature" srsName="http://www.open-
gis.net/def/crs/EPSG/0/4258" srsDimension="2">
<gml:name>Tarvasjoki</gml:name>
<gml:pos>60.58333 22.73333</gml:pos>
</gml:Point>
</gml:pointMembers>
</gml:MultiPoint>

```

```

</sams:shape>
</sams:SF_SpatialSamplingFeature>
</om:featureOfInterest>
<om:result>
<wml2:MeasurementTimeseries gml:id="mts-1-1-temperature">
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T01:00:00+03:00</wml2:time>
<wml2:value>7.93</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T02:00:00+03:00</wml2:time>
<wml2:value>6.98</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T03:00:00+03:00</wml2:time>
<wml2:value>5.82</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T04:00:00+03:00</wml2:time>
<wml2:value>5.57</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T05:00:00+03:00</wml2:time>
<wml2:value>5.69</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T06:00:00+03:00</wml2:time>
<wml2:value>6.2</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T07:00:00+03:00</wml2:time>
<wml2:value>8.7</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T08:00:00+03:00</wml2:time>
<wml2:value>10.92</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T09:00:00+03:00</wml2:time>
<wml2:value>13.72</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T10:00:00+03:00</wml2:time>
<wml2:value>16.37</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T11:00:00+03:00</wml2:time>
<wml2:value>18.12</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T12:00:00+03:00</wml2:time>
<wml2:value>18.95</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T13:00:00+03:00</wml2:time>
<wml2:value>19.38</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>

```



```

<wml2:MeasurementTVP>
<wml2:time>2016-05-22T14:00:00+03:00</wml2:time>
<wml2:value>19.71</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T15:00:00+03:00</wml2:time>
<wml2:value>19.48</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T16:00:00+03:00</wml2:time>
<wml2:value>19.52</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T17:00:00+03:00</wml2:time>
<wml2:value>19.56</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T18:00:00+03:00</wml2:time>
<wml2:value>19.0</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T19:00:00+03:00</wml2:time>
<wml2:value>18.35</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T20:00:00+03:00</wml2:time>
<wml2:value>17.16</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T21:00:00+03:00</wml2:time>
<wml2:value>15.84</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T22:00:00+03:00</wml2:time>
<wml2:value>13.81</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-22T23:00:00+03:00</wml2:time>
<wml2:value>11.45</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T00:00:00+03:00</wml2:time>
<wml2:value>9.84</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T01:00:00+03:00</wml2:time>
<wml2:value>8.9</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T02:00:00+03:00</wml2:time>
<wml2:value>8.05</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T03:00:00+03:00</wml2:time>
<wml2:value>7.21</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>

```

```
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T04:00:00+03:00</wml2:time>
<wml2:value>7.01</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T05:00:00+03:00</wml2:time>
<wml2:value>6.77</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T06:00:00+03:00</wml2:time>
<wml2:value>7.61</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T07:00:00+03:00</wml2:time>
<wml2:value>9.68</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T08:00:00+03:00</wml2:time>
<wml2:value>12.97</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T09:00:00+03:00</wml2:time>
<wml2:value>16.08</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T10:00:00+03:00</wml2:time>
<wml2:value>18.06</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T11:00:00+03:00</wml2:time>
<wml2:value>19.39</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
<wml2:point>
<wml2:MeasurementTVP>
<wml2:time>2016-05-23T12:00:00+03:00</wml2:time>
<wml2:value>20.38</wml2:value>
</wml2:MeasurementTVP>
</wml2:point>
</wml2:MeasurementTimeseries>
</om:result>
</omso:PointTimeSeriesObservation>
</wfs:member>
</wfs:FeatureCollection>
```