

Leo Vartio

# Kaukolämmönsiirraseman logiikkaohjelman takaisinmallinnus

Metropolia Ammattikorkeakoulu  
Insinööri (AMK)  
Automaatiotekniikka  
Insinöörityö  
29.11.2011

Tekijä(t) Otsikko Sivumäärä Aika	Leo Vartio Kaukolämmönsiirraseman logiikkaohjelman takaisinmallinnus 38 sivua + 6 liitettä 29.11.2011
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Automaatiotekniikka
Suuntautumisvaihtoehto	Prosessiautomaatio
Ohjaaja(t)	Yliopettaja Jouni Jokelainen Kunnossapitoinsinööri Timo Linna
<p>Työssä takaisinmallinnetaan Vantaan Energian ja Helsingin Energian kaukolämpöverkkojen välisen 80 MW:n lämmönsiirraseman vanha Siemensin Simatic S5 -pohjainen paikallisautomaatiojärjestelmä. Vanhan järjestelmän toiminnasta ei ole kattavaa kuvaa kellään Vantaan Energian palveluksessa nykyisin olevalla henkilöllä, sillä tieto on lukuisten päivitysten ja vanhan henkilöstön poistumisen myötä hävinnyt. Lisäksi Siemensin Simatic S5 -tuoteperhe on poistumassa lähivuosina markkinoilta, jolloin aseman käynnissäpidettävyys heikkenee varaosien heikon saatavuuden takia.</p> <p>Aseman Simatic S5 -logiikan ohjelmassa on käytössä noin 250 I/O-osoitetta, joista noin 50 on analogituloja ja -lähtöjä. Lisäksi ohjelmassa on käytössä pitkälti toistasataa merkkiä ja yhteensä noin 5000 riviä lähinnä AWL-muotoista ohjelmakoodia. Lisäksi Simatic S5 -logiikassa on kommunikaatioprosessori ja tämän toimintaa ohjaava oma ohjelmansa, joka tässä työssä on jätettiin toimenpiteiden ulkopuolelle.</p> <p>Takaisinmallintamisen tuloksena asemasta kirjoitettiin noin 100-sivuinen dokumentti, joka koostuu pääosin erilaisista ohjelman toimintaa selventävistä kaavioista, taulukoista ja näihin liittyvistä huomioista ja ohjeista. Lisäksi dokumentin alkuun sisältyy suppeahko toimintaselostus aseman perustoimintojen käyttämisestä.</p> <p>Asemalla todettiin takaisinmallintamisen myötä olevan tiettyjä puutteita ohjelmassa ja muutamia suoranaisia ohjelmointivirheitä. Ohjelmointivirheet tosin ovat onneksi sen luonteisia, ettei niistä normaalitilanteissa aiheudu välitöntä haittaa kaukolämmön jakelulle, mutta ne on uudelleensuunnittelun ja logiikan migraation myötä syytä korjata.</p>	
Avainsanat	automaatio, Simatic S5, PLC, takaisinmallinnus, käänteissuunnittelu, kaukolämpö

Author(s)	Leo Vartio
Title	Reverse engineering of the heat exchange stations PLC-program in district heating network.
Number of Pages	38 pages + 6 appendices
Date	29 Nov 2011
Degree	Bachelor of Engineering
Degree Programme	Automation Technology
Specialisation option	Process Automation
Instructor(s)	Timo Linna, Maintenance Engineer Jouni Jokelainen, Principal Lecturer
<p>The subject of this thesis is to reverse-engineer an old Siemens Simatic S5 PLC system in one 80 MW heat exchange station between district heating networks of Vantaan Energia Ltd. and Helen Group. The station is owned by energy company Vantaan Energia Ltd.</p> <p>The Vantaan Energia personnel did not have any extensive overview of the functionality of the old heat exchange station due to the many upgrades made to the station and old personnel leaving from the company. Above all else the Simatic S5 line of PLCs is coming to an end of the production so the maintenance would be a major problem in coming years if nothing is done to change the PLC platform to more modern.</p> <p>The station's old Simatic S5 PLC system does have circa 250 I/O addresses, of which 50 are analogy I/O addresses. The program of the Simatic S5 unit consists of about 5000 lines of code in STL format. On top of that there are more than 100 memory flags used in the program. The station is connected to the SCADA system with communication processor with its own program, the reverse engineering and analysis of which this thesis doesn't include.</p> <p>Concrete results of the reverse engineering is the document consisting of various types of diagrams, tables, flow diagrams and descriptions relating to these. Moreover, there is a description of the station's basic user routines which include start and stop procedures of the heat exchange process and the situations where the PLC runs to power limiting mode if some limit values are met. The document has circa 100 pages of information of how and why the station works like it does, this information is to help the re-engineering and migration project from old Simatic S5 PLC to some newer generation platform.</p> <p>The reverse engineering process revealed some major, but not fatal issues in PLC program. These issues include a few programming errors and some illogical construction in PLC program, but these are situated in program parts where these don't make fatal interruption to the district heating. Nevertheless these should be corrected in the re-engineering process before the migration.</p>	
Keywords	automation, Simatic S5, PLC, reverse engineering, re-engineering, district heating

## Sisällys

1	Johdanto	1
1.1	Vantaan Energia Oy	1
1.2	Kaukolämpö	2
1.3	Siemensin Simatic S5 -tuoteperheen historia ja tulevaisuus lyhyesti	3
2	Automaatiojärjestelmän ohjelma	3
2.1	Suljettu ja avoin tietokonejärjestelmä	3
2.2	Ohjelmien evoluutio	5
2.3	Uudelleensuunnittelu ja takaisinmallinnus	6
2.3.1	Uudelleensuunnittelu	6
2.3.2	Takaisinmallinnus	9
3	Kaukolämmönsiirrinasema B:n lähtötilanne	10
3.1	Ajossa oleva laitteisto	11
3.1.1	Siemens Simatic S5	11
3.1.2	Muu laitteisto lyhyesti	12
3.2	Käytettävissä oleva dokumentaatio	12
3.3	Käytettävissä oleva nk. yritysmuisti	13
3.4	Perinneojelma	13
3.5	Ohjelmallinen migraatio SIMATIC S5:stä S7:ään	15
4	LSB:n ohjelman takaisinmallintaminen	16
4.1	Työn aloitus	16
4.2	Ohjelmalliset apuvälineet työskentelyssä	17
4.2.1	Erot tiedostoissa	17
4.2.2	Heksaeditori	19
4.2.3	Taulukkolaskentaohjelma	19
4.3	Tietolähteet ja kritiikki	20
4.4	Alkutietojen kerääminen ja analysointi	21
4.5	Symbolitauluja vanhoista dokumenteista	21
4.6	Nykyisen ohjelman arviointi	22
4.7	Ohjelman takaisinmallintamisesta	23
4.8	Alkuvaiheen harha-askeleita	24
4.9	PI-kaavio	27

4.10	Löytyneiden tietojen tallentaminen	28
4.11	Logiikkaohjelmaa takaisinmallintaessa tehtyjä huomioita	30
4.11.1	Tehonrajoitus virtauksen mukaan	30
4.11.2	Virtauksen säätö	31
4.11.3	Säätäjän ohjausarvon muuntaminen aika-arvoksi	32
4.12	Takaisinmallintamisen avulla tuotettu uusi dokumentaatio	33
5	Yhteenveto	34
5.1	Ajatuksia migraatiota varten	35
5.2	Takaisinmallintamisen vahvuuksia	36
5.3	Takaisinmallintamisen heikkouksia	37
	Lähteet	38

## Liitteet

Liite 1. Heidehofintien lämmönsiirrinasema - LSB

Liite 2. LSB:n muutoshistoriaa aikajanan muodossa

Liite 3. LSB:n kl-veden virtausreitit, kun kaukolämmön siirtosuuntana on HE:lta VE:lle

Liite 4. LSB:n kl-veden virtausreitti, kun kaukolämmön siirtosuuntana on VE:lta HE:lle

Liite 5. Esimerkki LSB:n Simatic S5 –ohjelman tulkinnasta

Liite 6. LSB:n autostop-toiminto vuokaavioesityksenä

## Lyhenteitä ja käsitteitä

### Automaatiojärjestelmä

Kaikki tietyn prosessin tai tehtävän suorittamiseen liittyvä automaatio, myös fyysisesti muualla sijaitseva.

AWL	<i>Anweisungslist</i> . Simatic S5 -ohjelman käskylista-muotoinen esitystapa.
DB	Tiedostoyksikkö (Simatic S5).
ERP	<i>Enterprise Resource Planning</i> . Toiminnanohjausjärjestelmä.
FB	Toimintayksikkö (Simatic S5).
FBD	<i>Function Block Diagram</i> . Lohkokaavio-muotoinen esitystapa.
FC	FB:n kaltainen ohjelmanyksikkö Siemensin Step7-kielessä.
Funktio	Simatic S5:n aliohjelma, jonka suorituksen jälkeen varsinainen ohjelma jatkaa paikasta, johon kutsuttaessa ohjelman suoritus jäi.
HE	Helsingin Energia Oy.
IL	<i>Instructio List</i> . Käskylistaohjelmointi, mahdollisesti standardin IEC 61131-3 mukainen.
LSB	Lämmönsiirrinasema B.
Makro	Tietokoneohjelmien erityinen laji. Makroilla käyttäjä voi automatisoida ohjelman tai ohjelmiston toimintoja.
Merkkeri	Siemensin logiikkaohjelmien yhteydessä yleisesti käytettävä termi, jolla tarkoitetaan laitteen sisäisiä muuttujia, "apumuuttujia".
MES	<i>Manufacturing Execution System</i> . Tuotannonohjausjärjestelmä.

### Paikallisautomaatiojärjestelmä

Kaikki tietyssä fyysisessä sijainnissa oleva automaatiolaitteisto.

OB	Organisaatioyksikkö (Simatic S5).
PB	Ohjelmayksikkö (Simatic S5).
STL	<i>Statement List</i> . Simatic S5 -ohjelman käskylista-muotoinen esitystapa.
Vdc	<i>Volts-direct-current</i> . Tasajännite.
VE	Vantaan Energia Oy.

# 1 Johdanto

Tässä opinnäytetyössä oli tarkoituksena laatia Vantaan Energialle selvitys tämän LSB-kaukolämmönsiirrinaseman toiminnasta sen paikallisautomaation osalta. Kyseinen kaukolämmönsiirrinasema on nimellisteholtaan 80 MW, ja se toimii Helsingin Energian ja Vantaan Energian verkkojen välillä mahdollistaen kaukolämpöenergian siirtämisen verkkojen välillä.

Työn tulos on pohjustusta tulevaan automaatiojärjestelmän modernisointiin, sillä nykyinen järjestelmä on rakennettu perustuen Siemensin ohjelmoitavaan Simatic S5 -tuoteperheen logiikkaan, joka on poistumassa markkinoilta. Siksi tulevaa käynnissä- ja kunnossapitoa ajatellen on järjestelmä vaihdettava uuteen järjestelmään, jonka varosien saanti on turvattu myös muutaman vuoden kuluttua. Samalla työssä on tarkoituksena selvittää ja koostaa asemalle ylös- ja alasajorutiineita ja muita vastaavia, koska aseman toiminnasta ei ole ajantasaista dokumentaatiota eikä nykyinen kunnossapitohenkilöstökään ole täysin selvillä aseman toiminnasta, sillä aseman alkuperäiset vastuhenkilöt ovat poistuneet VE:n (Vantaan Energian) palveluksesta. Henkilöstön poistumisen yhteydessä aseman kirjattu ajantasainen dokumentaatio on kadonnut, mikäli sellaista on ollut olemassa.

Työtä tehdessä on myös pyrkimyksenä tarkastella PC-ohjelmoinnin puolella nykyisin jo varsin laajassa käytössä olevia takaisinmallintamisen ja uudelleensuunnittelun keinoja ja tarpeellisuutta automaation logiikkaohjelman toiminnan selvittämisessä (Wikipedia 2011b).

## 1.1 Vantaan Energia Oy

Työn tilaaja on Vantaan ja Helsingin kaupunkien omistama Vantaan Energia Oy. Omistus jakaantuu suhteessa 60/40 siten, että Vantaan kaupunki on pääosakas. Vantaan Energian historia alkaa vuodesta 1910, jolloin perustettiin Malmin Sähkölaitos Oy. Nyt sadan vuoden kuluttua ja lukuisten toiminimien jälkeen Vantaan Energiasta on kasvanut kansainvälinen energiakonserni. Vantaan Energia on yksi suurimmista suomalaisista energiayhtiöistä, ja sen liikevaihdon on 283,3 miljoonaa euroa (2010).

Vantaan Energian liiketoiminnot muodostuvat sähkön tuotannosta, sähkön siirrosta, kaukolämmön tuotannosta ja myynnistä sekä sähkön vähittäismyynnistä yksityis- ja yritysasiakkaille. (Vantaan Energia 2011)

## 1.2 Kaukolämpö

Kaukolämpö on Suomen suosituin lämmitysmuoto. Sen osuus on lähes 50 %, mikä on yli kaksinkertainen verrattuna seuraavaksi suosituimpaan sähkölämmitykseen.

Kaukolämmityksen keksijänä pidetään hollantilaista Cornelius Drebbeliä vuonna 1622. Vuonna 1877 Yhdysvalloissa näki päivänvalon ensimmäinen kaupallisesti toimiva kaukolämpöjärjestelmä keksijä Birdsill Hollyn toimesta. Suomessa ensimmäinen kaukolämmitetty kaupunginosa syntyi vuonna 1940 Helsingin Olympiakylään. Seuraavina vuosikymmeninä Suomessa kaukolämmitys kehittyi ja vakiinnutti asemansa taajamien lämmitysmuotona 1990-luvun loppuun tultaessa.

Kaukolämpö antaa hyvät mahdollisuudet päästöjen vähentämiselle. Tämä johtuu kaukolämmön tuotannon rakenteesta, jossa etenkin Suomessa käytetään hyväksi paljon teollisuuden ja sähköntuotannon hukkalämpöä. Näin menetellen tuotantolaitoksen kokonaishyötysuhde kasvaa merkittävästi. Suomi onkin 80 %:n osuudellaan johtava maa maailmassa, mitä tulee kaukolämmön tuottamiseen lämmön ja sähkön yhteistuotannolla. Suosimalla valtakunnallisesti jätteistä tuotettua kaukolämpöä, prosessien hukkalämpö mukaan luettuna, saavutetaan kansallisella tasolla päästöjen ja jätteiden määrän väheneminen sekä määrällisesti energiaraaka-aineiden tuonnissa merkittäviä säästöjä. Näiden yhteisvaikutuksena myös ympäristön kuormitus pienenee.

EU:ssa vuonna 2006 sähköntuotannosta oli vain 10 % lämmön ja sähkön yhteistuotantoa, kun vastaava osuus Suomessa oli kolmannes kaikesta tuotetusta sähköstä. (Kaukolämmön käsikirja 2006: 25 – 33)

Kaukolämpöä siirretään kaukolämpöverkossa, joka yleisesti muodostuu maanalaisista putkistoista ja tarpeellisissa paikoissa sijaitsevista kaukolämpöpumppaamoista, jotka huolehtivat tarvittavasta virtauksesta kaukolämpöverkossa. Tämän lisäksi kaukolämpö-

verkkoon kuuluu esimerkiksi lämmönsiirrinasemia, lämmöntuotantoyksiköitä ja tietenkin loppukäyttäjiä tarvittavine teknisine laitteistoineen.

### 1.3 Siemensin Simatic S5 -tuoteperheen historia ja tulevaisuus lyhyesti

Ensimmäiset tuotteet Siemensin Simatic S5 -tuoteperheestä tulivat markkinoille jo vuonna 1979. Se on siis ollut markkinoilla jo viitenä vuosikymmenenä, kokonaisena tuoteperheenäkin jo yli kaksi vuosikymmentä. Tämä erittäin pitkä markkinoillaoloaika on johtanut erittäin laajaan asennuskantaan maailmalla. On kuitenkin tultu tuotteen elinkaarensa vaiheeseen, jossa tuote on loppunut tuotannosta, mutta tuotteen saataavuus varaosina on vielä kuitenkin taattu Siemens Oy:n toimesta muutamaksi vuodeksi. Tämä suosittu ja laajassa käytössä olevan tuoteperheen poistuminen kokonaan markkinoilta johtanee tulevina vuosina suureen ohjelmoitavien logiikoiden uusimistarpeeseen teollisuudessa (Siemens 2011a; Siemens 2011b) – myös Suomessa.

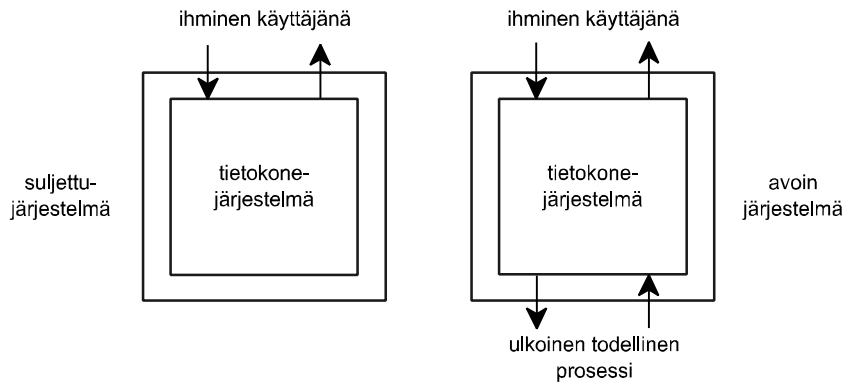
## 2 Automaatiojärjestelmän ohjelma

### 2.1 Suljettu ja avoin tietokonejärjestelmä

Tietokonejärjestelmät voidaan jakaa karkeasti kahteen luokkaan: avoimiin ja suljettuihin järjestelmiin. Jälkimmäisiin kuuluvat erilaiset esimerkiksi tieteellisessä laskennassa käytetyt järjestelmät, joissa järjestelmä kommunikoi ympäröivän todellisuuden kanssa lähinnä näppäimistön, hiiren ja näyttöpäätteen välityksellä eli ihmisen kanssa tai välityksellä. (Motus & Rodd 1994: 2 – 3)

Ensin mainittuun eli avoimeen järjestelmään luetaan kuuluvan esimerkiksi reaaliaikajärjestelmät, joihin automaationkin sovellukset kuuluvat. Avoin järjestelmä on siis järjestelmä, joka on vuorovaikutuksessa ympäröivän fyysisen maailman kanssa suoraan antureiden, toimilaitteiden tai muiden kulloinkin tarvittavien laitteiden välityksellä. (Motus & Rodd 1994: 2 – 3, Allworth & Zobel 1991: 1 – 2)

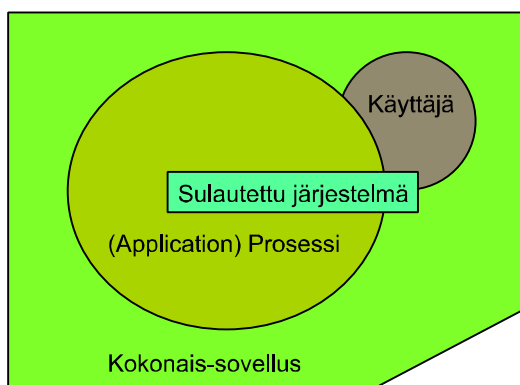
Tätä kahden järjestelmän eroa on havainnollistettu kuvassa 1, jossa on vasemmalla esitetty suljettu ja oikealla on avoin järjestelmä.



Kuva 1. Avoin ja suljettu tietokonejärjestelmä. (Motus & Rodd 1994: 3)

L. Motus ja M.G. Rodd sijoittavat automaatiojärjestelmät osaksi avoimia järjestelmiä, mutta samalla toteavat automaatiojärjestelmien luonnetta kuvaavan paremmin embedded (sulautettu) kuin real-time (tosi-aikainen) järjestelmä. Tätä perustellaan ohjelmistoalustojen kiinteällä sijoittamisella ohjattaviin järjestelmiin siten, että ne ovat ihmisen syötteiden kanssa kiinteänä osana isompaa kokonaisuutta eivätkä irtonaisena. Kuvassa 2 esitetään järjestelmän osat ja lomittuminen avoimessa järjestelmässä.

Sulautettu järjestelmä esimerkiksi kuvassa 2 koostuu ohjelmakoodista ja sen suorittamiseen tarvittavasta laitteistosta, se on siis ohjaava järjestelmä. Tämä kytkeytyy prosessiin tilanteen mukaisten laitteiden avulla. Kuvassa 2 olevan prosessin voidaan sanoa olevan ohjattava systeemi, jonka toimintaa sulautettu järjestelmä manipuloi vaikuttaen näin koko kuvan 2 kokonaisuuteen eli ympäristöön. (Allworth & Zobel 1991: 1)



Kuva 2. Sulautetun tietokonejärjestelmän sijoittuminen kokonaisuuteen. (Motus & Rodd 1994: 4)

Automaatiojärjestelmien ohjelmistot ovat olleet pitkään sidottuina laitevalmistajakoh-  
tasiin ohjelmointikieliin ja laitteisiin. Tämä on osaltaan haitannut automaatioalan oh-  
jelmistotyökalujen kehittymistä. Valmistajakohtaiset ohjelmointikieliset ja tavat ovat  
myös jatkuvasti hankaloittamassa kunnossapidon ponnisteluja sulautetuissa automaati-  
on ohjausjärjestelmissä, koska laitekannan uudistaminen on sidottu jo jopa vuosikym-  
meniä aikaisemmin tehtyihin laitevalintoihin. (Strömman 2002: 54) Tämä osaltaan lisää  
merkittävästi kunnossapidon kustannuksia, kun laitekannan vaihtaminen tulee ajankoh-  
taiseksi.

## 2.2 Ohjelmien evoluutio

On esitetty, että sulautetut ohjelmat vanhenevat ajan myötä niiden rakenteen moni-  
mutkaistuessa samalla, toisin sanoen niiden ohjelmallinen arkkitehtuuri rapistuu. Tästä  
rakenteen rapistumisesta johtuen myös kunnossapidon kustannukset nousevat jossain  
vaiheessa ohjelman elinkaarta niin suuriksi, ettei normaali kunnossapito ole enää kan-  
nattavaa. (Allworth & Zobel 1991: 6 – 8) Tämä ohjelmien vanheneminen noudattaa  
kuutta lakia, jotka Belady ja Lehman esittivät tutkimustensa perusteella kirjassaan  
vuonna 1976. Lisäksi Lehman on myöhemmin esittänyt kaksi lakia lisää entisen kuuden  
lisäksi. Seuraavassa on esitetty ensimmäinen ja toinen Lehmanin ohjelmien evoluu-  
tiolaeista (Allworth & Zobel 1991, Wikipedia 2011a):

1. Jatkuvan muutoksen laki – Käytetty ohjelma käy läpi jatkuvia muutoksia,  
kunnes se todetaan päivityskelvottomaksi.
2. Lisääntyvän monimutkaisuuden laki – Ohjelma käy aikaa myöten entistä  
monimutkaisemmaksi, mikäli erityistä huomiota ei kiinnitetä tämän estämiseen.

Ohjelmien elinkaaren kustannusjakauma näyttää noudattavan osin Lehmanin lakeja. Aluksi suunnittelu- ja implementointivaiheissa ohjelman kustannukset ovat korkeat, tämän jälkeen hyvin tasaiset hieman nousten, kunnes jossain vaiheessa ylläpitokustannukset rupeavat kohoamaan päätyen lopulta lähes yhtä kalliiksi kuin elinkaaren alkuvaiheessa. Tätä on havainnollistettu kuvassa 3.



Kuva 3. Ohjelman elinkaaren kustannuskehitys (Allworth & Zobel 1991: 7)

## 2.3 Uudelleensuunnittelu ja takaisinmallinnus

### 2.3.1 Uudelleensuunnittelu

Automaatiossa, kuten monen muunkin tekniikan alalla, on uudelleensuunnittelu varsin arkipäiväinen toimenpide. Kuitenkin tuntuu, ettei aiheesta juurikaan ole automaatioon liittyen kirjoitettu, niin vaikeata on löytää aiheesta kirjallisuutta. Liekö sitten synnä ajan- tai kirjallisuuden puute, ainakin kotimaista tietoa on vaikea löytää aiheesta. Tätä työtä tehdessä olenkin käyttänyt lukuisia tunteja yrittäen löytää automaatioon keskittyviä uudelleensuunnittelua tai takaisinmallinnusta käsitteleviä teoksia varsin laihoja tuloksia. Olenkin joutunut turvautumaan lähinnä englanninkieliseen kirjallisuuteen, joka korkeintaan sivuaa automaatiota.

Monesti käytännössä kunnollisen uudelleensuunnittelun sijasta päädytään tekemään nopeita ratkaisuja "padon paikkauksia" välittömien kustannuksien pakottamina tai ympäristötekijöiden vuoksi. Tällaiset ratkaisut johtavat usein pidemmällä aikavälillä vaikeisiin tilanteisiin, kun järjestelmän rakenne hämärtyy ja paikkaus "parsimalla reikä umpeen" jo itsessään aiheuttaa ongelmia toisaalla. (Demeyer ym. 2008.) Tämä toiminta toteuttaa luvussa 2.2 esitettyä toista Lehmanin ohjelmien evoluutiolakia. Lisäksi jos

järjestelmän toiminnan ymmärrys on jäänyt pinnalliseksi muutoksia tai korjauksia tehdessä, aiheutuu helposti uusia ongelmia jatkossa, jolloin kunnossapidon kustannukset nousevat entisestään. (Demeyer ym. 2008)

Kirjallisuudessa on vähemmän käsitelty konkreettisella tasolla automaation uudelleensuunnittelua ja käänteismallintamista, mutta esimerkiksi tietokanta- ja olio-ohjelmointimaailmassa on asiasta kirjoitettu jo 1990-luvun alusta lähtien. Näistä kirjoituksista ja kirjoista osa on varsin yleispäteviä ja ne antavat yleisemminkin näkemystä systeemien uudelleensuunnitteluun (Demeyer ym. 2008: 6). Seuraavaan olen lainannut muutamia syitä ja indikaattoreita takaisinmallintamisen ja uudelleensuunnittelun tarpeesta teoksesta Object-oriented reengineering patterns, jossa kirjan tekijät itsekin mainitsevat, että monet kirjan tekniikoista sopivat kaikkiin ohjelmallisiin järjestelmiin, vaikkakin ovat erinäisistä syistä kohdistaneet huomionsa olio-ohjelmointiin. Mielestäni seuraavat kirjassa esitetyt syyt ja indikaattorit ovat niin yleispäteviä, että sopivat automaatiojärjestelmiinkin. Nämä lainaukset olen tulkinut suomeksi kunkin lainauksen alle *kursiivilla*. Samalla on mainittava, että LSB-lämmönsiirraseman paikallisautomaation ohjelmassa toteutuvat kaikki alla esitetyt merkit ja syyt takaisinmallintamiseen ja uudelleensuunnitteluun.

Limited understanding of the entire system. Not only does nobody understand the fine print, but hardly anyone has a good overview of the whole system.

*Koko järjestelmän toiminnan ja toimintojen ymmärrys on rajoittunut. Järjestelmän parissa työskentelevät eivät tunne pieniä yksityiskohtia, mutta kenelläkään ei myöskään ole kokonaiskuvaa koko järjestelmästä ja siitä, mikä siihen liittyy.*

Maintenance Dependencies. When you fix a bug in one place, another bug pops up somewhere else. This is often a sign that the architecture has deteriorated to the point where logically separate components of the system are no longer independent.

*Riippuvaisuus kunnossapidosta. Kun virhetoiminto yhdessä paikassa korjataan, niin kohta toisaalla ilmenee uusi ongelma, joka vaatii korjausta. Tämä on usein merkki järjestelmän arkkitehtuurin heikkenemisestä, jolloin sen loogisesti irralliset komponentit eivät enää ole itsenäisiä soluja.*

On huomioitava esimerkiksi ala-asemalle tehtyjen pienten päivitysten säteileminen kaukokäyttöjärjestelmän puolelle ja vastaavasti kaukokäytön muutoksen säteileminen ala-asemalle.

Automaatiojärjestelmissä on kuitenkin huomattava, ettei kuluvia fyysisiä osia voi laskea tähän, ellei sitten tilanne ole sellainen, että uuden osan asentaminen jonnekin aiheuttaa jonkin muun osan rikkoontumisen jossain muussa systeemin osassa ennenaikaisesti. Tästä päästäänkin tämän ideamallin laajentamiseen varsinaisen ohjelmallisen automaation ulkopuolelle.

Duplicated Code. Duplicated code arises naturally as a system evolves, as shortcut to implementing nearly identical code, or merging different versions of a software systems. If the duplicated code is not eliminated by refactoring the common parts into suitable abstractions, maintenance quickly becomes a nightmare as the same code has to be fixed in many places.

*Toistuvia koodirakenteita. Toistuvaa koodia kertyy luonnollisesti järjestelmän kehittyessä, kun oikaistaan samankaltaisten toimintojen implementoinnissa tai kun järjestelmiä yhdistetään. Jos toistuvia koodirakenteita ei refaktoroinnin avulla eliminoida sopivaan abstraktioon, niin järjestelmän ylläpito muodostuu nopeasti tuskalliseksi, kun koodia täytyy muokata useassa paikassa.*

Inside knowledge about system has disappeared. This is a bad sign. The documentation is out of sync with the existing code base. Nobody really knows how it works.

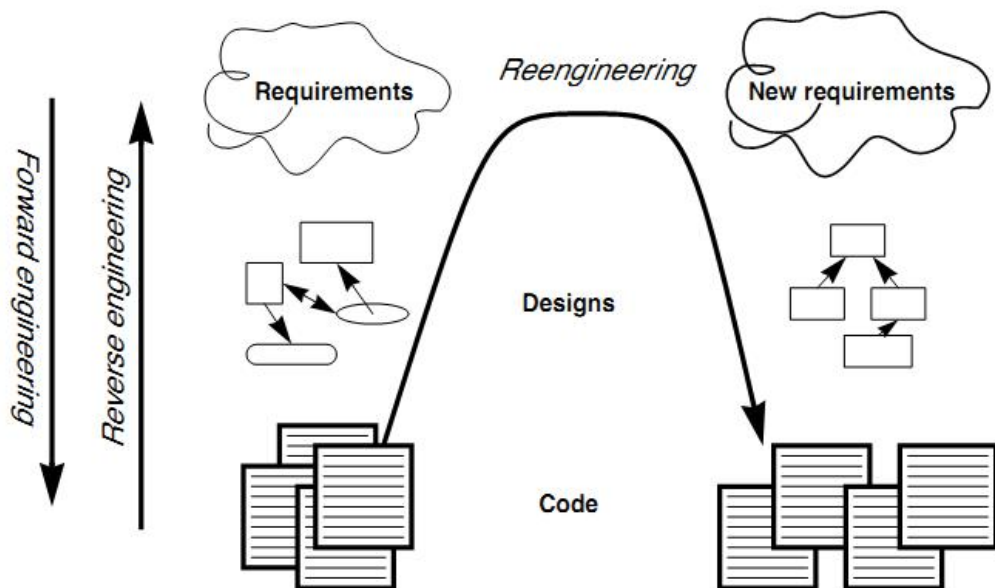
*Sisäinen näkemys systeemin toiminnasta on hävinnyt. Tämä on paha merkki. Dokumentaatio on jäänyt jälkeen olemassa olevasta systeemistä. Kukaan ei todellisuudessa tiedä, kuinka se toimii.*

Obsolete or no documentation. Obsolete documentation is a clear sign of a legacy system that has undergone many changes. Absence of documentation is a warning sign that problems are on the horizon, as soon as the original developers leave the project.

*Vanhentunut tai olematon dokumentaatio. Vanhentunut dokumentaatio on selvä merkki perinnejärjestelmän läpikäymistä lukuisista muutoksista. Vaillinainen dokumentaatio on selvä varoitussignaali tuloillaan olevista ongelmista, jotka syntyvät heti alkuperäinen henkilön lähtiessä projektista.*

### 2.3.2 Takaisinmallinnus

Takaisinmallinnus, josta käytetään myös nimityksiä käännteissuunnittelu tai käännteismallintaminen, on toimenpide, jolle ei ilmeisesti ole vakiintunutta suomalaista vastinetta, mutta usein suomessakin nykytavan mukaan käytetään englanninkielistä termiä *reverse engineering*. Se on insinöörien yksi monista perustyökaluista, joskin sen merkityksen ja sisällön pohdinta ja kehittäminen on ilmeisen pysähtynyttä. Tämä termi tarkoittaa toimenpiteitä, joilla pyritään analysoimalla, kokeilemalla ja tietoa keräämällä ymmärtämään jo olemassa olevaa vanhaa järjestelmää. Sen ei kuitenkaan tarvitse välttämättä olla ohjelmallinen, vaan järjestelmä voi olla muutakin. Valitettavasti tätä toimenpidettä ja uudelleensuunnittelun tärkeää apuvälinettä ei vaikeasti löydettävän kirjallisuuden valossa juurikaan ilmeisesti käytetä automaation yhteydessä. Kuvassa 4 on pyritty havainnollistamaan takaisinmallintamisen sijoittumista uudelleensuunnittelu-projektissa. Kuvasta käy ilmi toimenpiteen korostunut merkitys.



Kuva 4. Eteenpäin-, käänteis- ja uudelleensuunnittelu. (Demeyer ym. 2008: 9)

Takaisinmallinnuksen merkitys avoimissa ohjelmistojärjestelmissä, joihin automaationkin sovellukset kuuluvat, tulee vain korostumaan tulevina vuosina, kun vanhaa laitekantaa joudutaan päivittämään uusille alustoille. Myös suurta hyötyä takaisinmallintamisesta on prosessien dokumentaation uudelleenkoostamisessa, sillä on yleinen totuus, että vanhojen järjestelmien dokumentaatio haalistuu vuosien kuluessa, ts. dokumentaatio jää jälkeen itse prosessin tilanteesta, ja jossain vaiheessa vanha dokumentaatio

muuttuu lähes käyttökelvottomaksi tai ainakin luotettavuudeltaan kyseenalaiseksi, kun versioiden ero kasvaa tarpeeksi suureksi. (Demeyer ym. 2008)

Takaisinmallinnuksen merkitys on sen keskeinen merkitys uudelleensuunnittelun pohjustuksena tai dokumentaation uudelleen koostamisessa.

Automaation sovellusten yhteydessä takaisinmallintamista on käsitelty varsin niukasti ja jo mainittukin, että suomenkielisen kirjallisuuden löytyminen aiheesta on varsin niukkaa.

### 3 Kaukolämmönsiirrinasema B:n lähtötilanne

Tässä luvussa käsittelen LSB:llä käytössä olevaa paikallista automaatiolaitteistoa ja instrumentointia sekä itse prosessia ja sen laitteistoa siltä osin ja siinä laajuudessa, kuin sen tämän dokumentin ymmärtämiseksi katson olevan tarpeen. Rajaan siis kaukokäytön laitteiston, ohjelman ja sen toiminnan suurelta osin ulos selostuksesta. Lisäksi kuvailen LSB:stä käytettävissä olevaa dokumentaatiota, ajossa olevaa ohjelmaa sekä ohjelman edellisiä versioita.

LSB:kin on käynyt historiansa aikana lähes jatkuvaa muutosta, osa siitä on dokumentoitu, osa on jäänyt syystä tai toisesta dokumentoimatta tai on dokumentoitu vain osittain. Tämä on suurin yksittäinen haaste LSB:n automaatiojärjestelmän uudistamiseen tähtäävässä työssä. Tällainen tilanne on esitetty luvussa 2.3.1 yhdeksi indikaattoriksi uudelleensuunnittelun tai käännteissuunnittelun tarpeesta.

Lämmönsiirrinasema B on toteutettu tavalla, joka mahdollistaa kaikkien tilatietojen välittämisen myös HE:n tarkkailevaan järjestelmään.

Aseman silloinen käyttöhenkilöstö on tehnyt kirjattuja muutoksia ohjelmaan ainakin vuosina 1998 ja 1999, todennäköisesti jo aikaisemminkin, mutta näistä ei ole säilynyt mainintoja. Tällöin on muutettu aseman säätöalgoritmien arvoja, esimerkiksi säätöventtiilin 14 ohjauksessa käytettävää aikapulssia on muutettu kymmenestä sekunnista puoleen minuuttiin, jolloin vanha erosuure tallennetaankin aina joka 30:s sekunti.

Vuosina 2000 ja 2001 asemalla on tehty enemmän muutoksia tarkoituksena modernisoida asemaa. Mallina tässä on käytetty Savion lämmönsiirraseman rakenteita ja ohjelmaa. Tällöin asemalle on mm. asennettu erillinen paineenpitojärjestelmä, joka tasaa painetta ja nestemäärää kaukolämpöverkkojen välillä. Asemalle asennettiin ilmeisestikin tässä vaiheessa myös uusi pumppu Helsingin Energian kaukolämpöverkon puolelle, mikä mahdollistaa kaukolämmön myynnin molemmille osapuolille. Suurimpana uudistuksena oli kuitenkin aseman kytkeminen kaukokäytön piiriin, mihin on ilmeisesti otettu mallia myös Savion asemalta. Syystä tai toisesta ei näistä muutoksista ole käytettävissä kunnollista dokumentaatiota, syitä voi vain arvailla. Lisäksi silloinen henkilökunta on tavoittamattomissa.

Ainakin vuodesta 2005 on ohjelma pysynyt muuttumattomana, tällöin on ladattu viimeksi ohjelma logiikasta ohjelmointilaitteelle. Tämän ja nykyisen version vertaaminen ohjelmallisesti paljasti, ettei ohjelmissa ole eroja. (Linna 2011.)

### 3.1 Ajossa oleva laitteisto

Aseman laitteistoon kuuluu itsenäinen rakennusautomaatiojärjestelmä, kaukolämpöverkkojen välille asennettu itsenäinen nk. paineenpitojärjestelmä, HE:n kaukokäyttöön kytketty tarkkailujärjestelmä ja itse varsinainen kaukolämmön siirtimen ohjausjärjestelmä, johon edellä mainitut ainakin osin on liitetty, mutta joiden tietoa ei käytetä normaalissa ohjauksessa ja säädössä, vaan välitetään valituilta osin kaukokäytön valvomon.

#### 3.1.1 Siemens Simatic S5

LSB:llä on asennettuna seuraavat Simatic S5 -laitteet:

- 6ES5928-3UA12 – prosessori S5-135U (928A)
- 6ES5524-3UA13 – kommunikaatioprosessori (modeemiyhteys)
- 6ES5465-4UA12 – analogitulokortti, 16 kanavaa, 4 – 20 mA
- 6ES5465-4UA12 – analogitulokortti, 16 kanavaa, 4 – 20 mA
- 6ES5465-4UA12 – analogitulokortti, 16 kanavaa, 4 – 20 mA
- 6ES5470-4UC12 – analogilähtökortti, 8 kanavaa, 4 – 20 mA
- 6ES5431-4UA12 – digitaalitulokortti, 16 kanavaa, 24 Vdc

- 6ES5431-4UA12 – digitaalitulokortti, 16 kanavaa, 24 Vdc
- 6ES5431-4UA12 – digitaalitulokortti, 16 kanavaa, 24 Vdc
- 6ES5430-4UA14 – digitaalitulokortti, 32 kanavaa, 24 Vdc
- 6ES5430-4UA14 – digitaalitulokortti, 32 kanavaa, 24 Vdc
- 6ES5451-4UA13 – digitaalilähtökortti, 32 kanavaa, 24 Vdc
- 6ES5451-4UA13 – digitaalilähtökortti, 32 kanavaa, 24 Vdc
- 6ES5188-3UA31 – keskusyksikkö, 24 Vdc
- Virransyöttöyksikkö

Tulo- ja lähtökanavia on yhteensä 232 kappaletta. Analogisia kanavia edellä mainituista on 56 kappaletta, joista käytössä on 50 kappaletta.

### 3.1.2 Muu laitteisto lyhyesti

Asemalla käytössä olevat venttiilit ovat pääosin Auman käsi- ja moottoriventtiileitä. Kaukolämpöveden virtauksen tehostamisesta huolehtii neljä pumppua, joista kolme on tasajännitekäyttöjä. Mittauksina asemalla on paine-, lämpö-, virtaus-, asento- ja paineeromittauksia. Lisäksi asemalla on lukuisia tilatietoja välittäviä antureita lähinnä venttiileiden yhteydessä. Tarkemman kuvan aseman laitteistosta saa liitteenä 1 olevasta PI-kaaviosta.

### 3.2 Käytettävissä oleva dokumentaatio

LSB:n automaatiolaitteistosta on olemassa kaksi mappia sekalaisia dokumentteja, jotka suurimmalta osin osoittautuivat lähemmässä tarkastelussa ainakin osittain vanhentuneiksi.

Dokumenttien joukossa oli mm.

- sähkökuvat aseman säätöventtiileiden kytkennöistä
- varsinaiset sähkökuvat
- automaatiokaapin kalustuspiirustukset
- logiikan tulojen ja lähtöjen johdotukset
- automaatiokaapin ristiinkytkentäkuvat
- elektromekaanisen ohjaustaulun kytkentäkuvat ja ristiinkytkentätaulukot
- nk. paineenpitopiirin sähkökuvat ja ristiinkytkentätaulukot

- muutama vedos vanhoista Simatic S5 -ohjelmista
- I/O-listaus.

Tämän lisäksi löytyi vielä sähköisenä muutamia dokumentteja ja ohjelmia. Löytyneiden sähköisten dokumenttien hyöty selvitystyötä tehdessä jäi melko vähäiseksi, koska osoittautui, että nämäkin olivat kaikki jollain tavoin vanhentuneita, joten niiden luotettavuus oli kyseenalaistettava myös niiltä osin, joilta kaikki näytti muuttumattomalta.

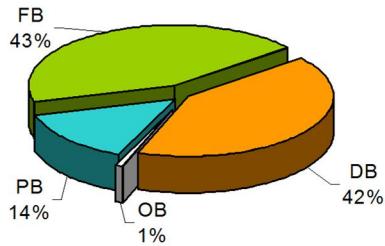
### 3.3 Käytettävissä oleva nk. yritysmuisti

Yksi keskeisiä LSB:n ongelmien syitä on tiedon puute, joka on aiheutunut LSB:n kanssa aikaisemmin työskennelleiden henkilöiden siirtymisestä tavoittamattomiin. Lisäksi tästä johtuen ajantasainen dokumentaatio on hävinnyt, mikäli sellaista on ollut.

Osittaista tietämystä aseman toiminnasta ja toiminnoista on toki edelleen käytettävissä, mutta kokonaiskuva aseman toiminnasta tai toimimattomuudesta on puutteellinen.

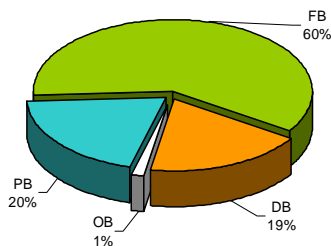
### 3.4 Perinneohjelma

LSB:n säädöstä ja ohjauksesta pääosin huolehtiva logiikkaohjelma on rakenteeltaan tyyppinen Simatic S5 135U -ohjelma. Se sisältää organisaatioyksiköitä (OB), ohjelmayksiköitä (PB), toimintayksiköitä (FB) ja tiedostoyksiköitä (DB) (Siemens Oy 1989: luku 1, 20). Näitä ohjelmanyksiköitä on kaikkiaan 32 kappaletta, joista kaksi on jätetty viimeisimmässä versiossa käyttämättä. Koodia näissä 32 yksikössä on kaikkiaan noin 5000 riviä. Ohjelma on pääosin kirjoitettu AWL-muodossa. Tämä on saksalaisen Step5-notaation mukainen lyhenne, sillä Suomessa on ollut käytössä paljolti saksalainen notaatio. Englanninkielisessä notaatiossa on AWL:ää vastaava lyhenne STL (Statement List). Tämä Step5 AWL -koodi on muodoltaan hieman vastaavaa kuin IEC 61131-3:n mukainen käskylistaohjelmointi (Instruction list), jonka lyhenne on IL (John & Tiegelkamp 2001: 95 – 97). Käskylistaohjelmoinnin suomenkielinen nimitys ei ole vakiintunut (Strömman 2002). Tämän tyyppinen ohjelmakoodi on lähellä assembly-tyyppistä ohjelmakoodia. Kuvassa 5 on esitetty ohjelmakoodin jakaantuminen eri lohkokotyyppien välille.



Kuva 5. Ohjelmakoodin likimääräinen jakaantuminen eri lohkojen kesken, kun käyttämättömät lohkot sisältyvät.

Ohjelmassa on kuitenkin jätetty käyttämättä kaksi lohkoa. Näistä toinen on suurehko tiedostolohko, joka sisältää ilmeisesti alkuperäiset tulojen ja lähtöjen symboliset nimet. Tätä on kaikkiaan noin 1500 riviä. Näin suurella rivimäärällä on jo varsin merkittävä vaikutus koodin jakaumaan eri lohkojen välillä, kuten kuvassa 6 on nähtävissä. Toinen käyttämättä jätetty lohko on toimintayksikkö sisältäen ainoastaan muutamia kymmeniä rivejä vaikuttaen varsin vähän prosentuaaliseen jakaumaan.



Kuva 6. Ohjelmakoodin likimääräinen jakautuminen eri lohkojen kesken, kun käyttämättömät lohkot on jätetty huomiotta.

Ohjelma on osittain aikariippuvainen toimintayksiköihin kirjoitetun ohjelman osalta, sillä niitä kutsutaan OB13:n kautta, joka on Siemens Simatic S5-135U -prosessorissa 100 ms:n välein ajettava organisaatioyksikkö. Kaikkiaan lähes puolet ohjelman koodiriveistä sijoittuu tähän 100 ms:n välein ajettavaan ohjelman osaan.

Ensimmäisenä sain tilaajalta käyttöni vuonna 2010 kertaalleen Converting S5 Files -ohjelman läpiajetun ohjelmaversiosta, joka oli käännetty vuoden 2002 ohjelmaversiosta. Tämä on puhdas tekstimuotoinen lähdekoodi, jonka alussa on mainittuna muutamia tärkeitä huomioita alustojen eroista Simatic S5:n ja S7:n välillä. Sovittuamme työstä paremmin sain käyttöni myös vanhat Simatic S5 -muotoiset ohjelmaversiot. Ne oli

päivätty kansioihin seuraavasti: 15.1.2001, 27.1.2001, 28.3.2001 ja 17.9.2002. Näistä 28.3.2001 oli tuorein kommentoitu ohjelmaversio. Kuitenkin oli tiedossa, että silloinen henkilö, joka asemaa oli hoitanut, oli jatkuvasti tehnyt pieniä muutoksia ohjelmaan (Linna 2011), ja tämä varmistuikin myöhemmin vertaamalla ohjelmaversioita keskenään.

### 3.5 Ohjelmallinen migraatio SIMATIC S5:stä S7:ään

Siemens on tiedostanut migraatiotarpeen Simatic S5 -pohjaisista järjestelmistä S7-järjestelmään ja toimittaa Simatic Manager -ohjelmiston mukana työkalua, jolla tämä on toteutettavissa. Tässä toimenpiteessä on kuitenkin ongelmia, joista kerron hieman seuraavassa.

Logiikan migraatio on mahdollista karkeasti kahdella tavalla, joko täysin käsityönä tai tukeutuen ohjelmalliseen ohjelman muunnokseen, joista jälkimmäinenkin tarkoittaa useimmissa tapauksissa lähes käsityönä toteutettavaa muunnosta. Ohjelmallisen muunnoksen Simatic S5:stä S7:ään mahdollistaa Siemensin Step7-ohjelmointiympäristön mukana toimitettava Converting S5 Files -työkalu, joka muuntaa ohjelman mahdollisuuksien mukaan käsky käskyltä uuteen S7-muotoon.

Suurimpana ongelmana tässä lähestymistavassa on, ettei täydellistä muunnosta ole mahdollista toteuttaa ohjelmallisesti. Tämä johtuu rakenteellisista ja muodollisista eroista järjestelmien välillä. Tämän takia jokainen muunnettava ohjelma tulee huolellisesti analysoida ja testata muunnoksen jälkeen.

Muunnoksessa syntyy usein virheitä, joten on helpointa luoda uusi S7-lohko, joka toteuttaa ulkoisesti Simatic S5 -lohkon toiminnon. (Brisky 2007: 12 – 15)

Tällainen todellisuudessa käsityönä suoritettava migraatio aiheuttaa tarpeen ymmärtää ohjelman toimintaa, ja tähänhän myös käänteismallintamisella yleensä pyritään.

## 4 LSB:n ohjelman takaisinmallintaminen

LSB:n tilanne on luvussa Uudelleensuunnittelu ja takaisinmallintaminen esitettyjen kohtien mukainen, ja nämä kohdat paljastavat LSB:n ohjelman vaativan uudelleensuunnittelua. LSB:n toimintaa ei kukaan nykyisistä henkilöistä täysin ymmärrä. Sen dokumentaatio on vähintäänkin puutteellinen, sen sisäistä rakennetta ei tiedetä ja koodissa on paljon toistoa.

Työssäni olen pyrkinyt selventämään LSB:n toimintaa ja suurelta osin myös onnistunut siinä. Tosin katson, etten ole täysin yksiselitteisesti onnistunut mallintamaan logiikkaohjelman rakennetta ja toimintaa todella pelkistetyksi esitykseksi. Tuottamani uusi dokumentaatio ohjelman toiminnasta vaatii jonkinasteista syventymistä ja automaation perusteiden ymmärtämistä.

### 4.1 Työn aloitus

Takaisinmallintamisesta ja uudelleensuunnittelusta toki löytyy paljon samaa kuin uudenkin suunnittelussa, ja lopputuloshan on ainakin teoriassa samanlainen, mutta analyysia hyvistä toimiviksi todetuista työskentelytavoista, apuvälineistä ja ohjelmista ei käytännössä ole kirjoitettu, kuten on jo mainittu luvussa 2. Automaation ohjelmien suunnittelun yhteydessä esimerkiksi Strömman (2002) antaa ymmärtää, ettei laaja yhtenäinen suunnittelumalli tai systeemi ole mahdollinen johtuen ohjelmistoalustojen (logiikoiden) perinteisestä yhteensopimattomuudesta kielen ja ohjelmistorakenteen osalta.

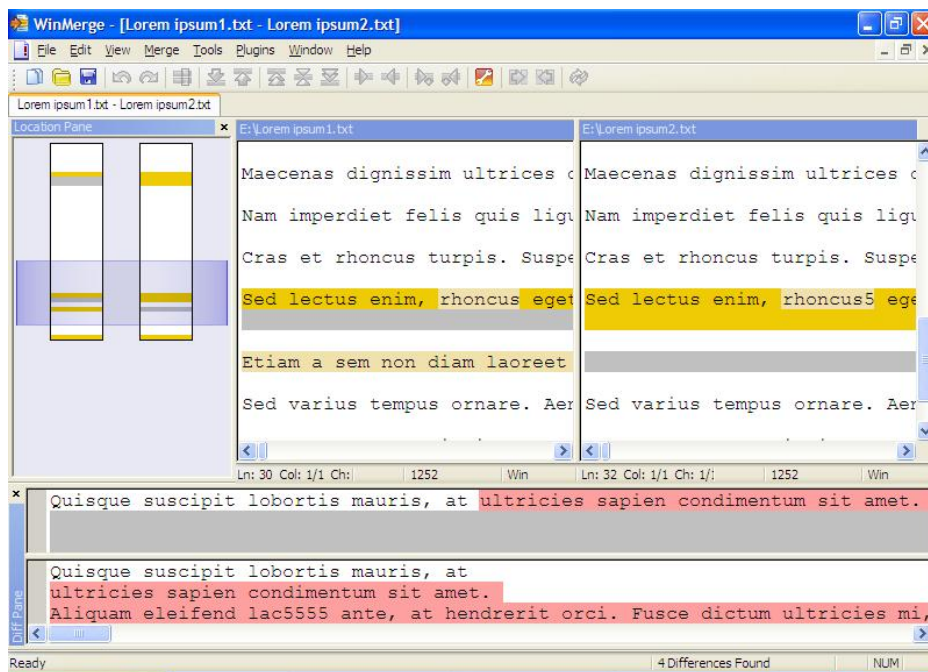
Varsinkin vanhojen perinneohjelmien ja järjestelmien osalta olisi kuitenkin todella ajankohtaista ja suorastaan kiireellistä perehtyä ja pyrkiä luomaan edes jotenkin yleistettävä toimintamalli takaisinmallinnuksen ja uudelleensuunnittelun osalta, sillä muutamassa vuodessa tulevat esimerkiksi Siemens Simatic S5 –pohjaiset järjestelmät elinkaarensa päähän, mikä aiheuttaa suomessakin valtavan järjestelmien modernisointi tarpeen.

## 4.2 Ohjelmalliset apuvälineet työskentelyssä

### 4.2.1 Erot tiedostoissa

Sähköisinä versioina löytyi mm. IO-lista kahtena versiona, ja näiden vertailun tein WinMerge-ohjelmalla, joka on tarkoitettu merkkimuotoisten tiedostojen vertailuun.

Muutamasta asemasta löytyneet dokumentit olivat päällisin puolin samankaltaisia ja kohtalaisen pitkiä. Toisaalta oli saatava selville, onko dokumenteissa tapahtunut muutoksia, ja jos on, niin miltä osin. Tähän oli oivana välineenä WinMerge-tietokoneohjelma, joka on tehty tekstidokumenttien vertailuun keskenään (kuvassa 7). Se näyttää värein muutetut kohdat ja rivit sekä eräänlaisena graafina dokumentin kokonaisuudessaan ja kaikki keskenään eroavat kohdat. Se on todella kätevä myös varsinaisen lähdekoodin vertaamiseen eri versioiden välillä, mikäli nämä ovat saatavilla tekstimuotoisina tiedostoina.



Kuva 7. WinMerge-ohjelman näkymä tiedostojen sisältöä vertaillessa.

Kuitenkin esimerkiksi IBH Softecin S5/S7 for Windows -ohjelman oma vertailutyökalu on huomattavasti kehittyneempi ja parempi, kun vertailun kohteena on esimerkiksi kaksi Simatic S5 -ohjelman versiota, sillä tämä S5/S7 for Windows on tehty Siemensin S5 ja S7 logiikoiden ohjelmointia varten. Tästä syystä ohjelman toiminnot on toteutettu

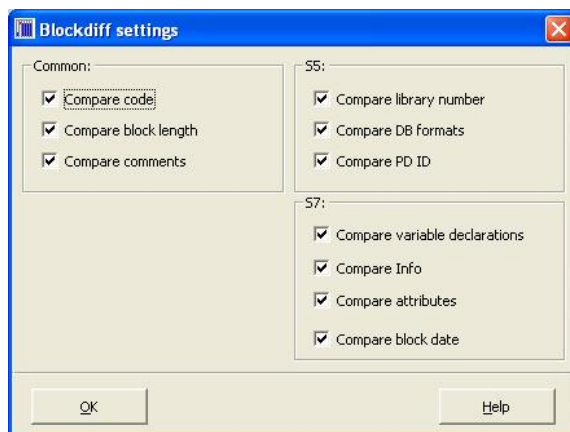
näitä kahta logiikkaa tukeviksi ja ohjelma osaakin näyttää suoraan, minkätyyppisestä muutoksesta ohjelmassa on kyse. Onko esimerkiksi koodi muuttunut, ovatko arvot muuttuneet, onko lohkon koko muuttunut jne.? Lisäksi ohjelma näyttää ohjelmaversioiden välisen koodin erot esimerkiksi punaisella (kuva 8). Lisäksi ohjelmassa voidaan määrittellä, mitä tietoja otetaan mukaan vertailuun, ja mm. pelkkien kommenttien vertailu on mahdollista. Vertailtavien tietojen valintaruutu on esitetty kuvassa 9.

FB 60	different	Differences in code
FB 70	different	Differences in code
FB 130	different	Differences in code
DB 10	different	Difference in actual values
DB 12	different	Difference in actual values
DB 13	different	Difference in actual values
DB 20	different	Difference in actual values
DB 40	different	Difference in actual values
FB 14	different	Differences in source code
FB 30	different	Differences in source code
DB 14	different	Differences in source code
OB 1	identical	

FB 70 in 160511ST	FB 70 in E:\LSB source\yms\LSB Source\28.3.2001\LSB@@@ST.55D
PG-ID: 84	PG-ID: 84
Bib-NR: 000000	Bib-NR: 000000
>>> Len: 100	>>> Len: 128
BEGIN	BEGIN
NETWORK 1	NETWORK 1
NAME: P4 HE>VE	NAME: P4 HE>VE
U A 8.7	U A 8.7
SPB =M001	SPB =M001
A DB 10	A DB 10
L DD 40	L DD 40
L KG 90	L KG 90
>G	>G
>>> = M 45.5	>>> S M 45.2
>>> U M 45.5	M001: ***
>>> L KT 300.2	
>>> SE T 9	NETWORK 2
>>> NOP 0	U A 8.7
>>> NOP n	n

Kuva 8. IBH Softecin S5 S7 for Windows compare-työkalun näyttämät erityyppiset erot LSB:n koodissa.

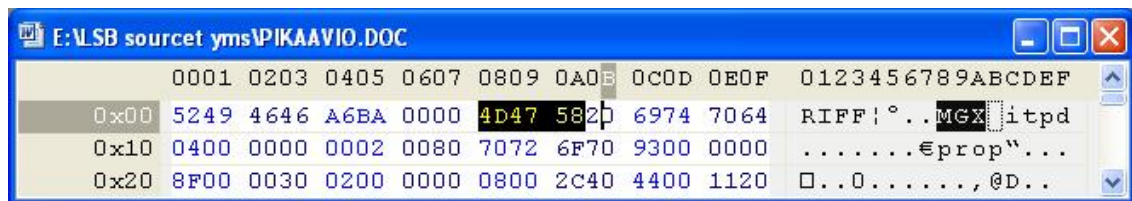


Kuva 9. IBH Softecin S5 S7 for Windows –ohjelman compare-työkalun asetusten valinta.

#### 4.2.2 Heksaeditori

Työssä vastaantulevien vanhojen dokumenttien alkuperäisen ohjelman selvittämisessä saattaa tulla erittäin hyödylliseksi nk. heksaeditori. Se on siis ohjelma, jolla voidaan tarkastella binäärimuotoisia tiedostoja ja muuttaa niiden arvoja tavuittain.

Eräs työn tilaajan sähköisistä arkistoista löytynyt tiedosto oli esimerkiksi nimetty PI-KAAVIO.DOC. Kun tätä tiedostoa yritti avata MS Wordissä, jonka tiedostomuoto doc yleisesti käsitetään, valitti Word, ettei tiedosto ei ole sovelias Word-asiakirja. Kun avasi tiedoston heksaeditoriin, kuten kuvassa 10 on nähtävissä, näkyi ensimmäisinä asioina eräs kolmikirjaiminen yhdistelmä. Internetistä hakemalla "Filetype mgx" tuli tieto kyseisen tiedostopäätteen kuuluneen Micrografx-ohjelmalle. Tämä tieto vahvisti, ettei kyseessä ole MS Word -asiakirja vaan väärin nimetty grafiikkatiedosto. Lisäksi olin törmännyt micrograph-nimeen keskusteluissa työnohjaajan Timo Linnan kanssa, joten tiedostoa yritettiin avata kyseisellä ohjelmalla, jolloin sieltä löytyi tiedostonimen mukainen PI-kaavio LSB-asemasta, joskin tämä oli osin vanhentunut.



Kuva 10. Esimerkki tiedoston PIKAAVIO.DOC alusta löytyvästä tiedostoformin lyhenteestä, kun tiedostoa katsotaan heksaeditorissa.

#### 4.2.3 Taulukkolaskentaohjelma

Excel-tietojen käsittelemiseen tein muutaman yksinkertaisen makron, joiden avulla sain vain tarpeelliset tiedot irrotettua useamman sarakkeen listoista. Hyödyllisiksi osoittautuivat etenkin makrot, joilla sain leikattua koko sarakkeessa olevista muuttujien nimistä vain tietyn mustatun alueen mittaisen merkkijonon, ja makro, joka kirjoittaa sarakkeeseen määrättyllä välillä bittiosoitteet, esim. 4.0, 4.1, ... , 5.5, 5.6 ja 5.7.

### 4.3 Tietolähteet ja kritiikki

Työssä korostuu hyvin vahvana lähdekritiikki. Kun ohjelmaa rupeaa selvittämään, on aluksi hyvin epäselvä kuva kohteen toiminnasta. Tätä kuvaa on vahvistettava ja lopuksi ruvettava värittämään. Ongelmaksi muodostuvat tietolähteet, tieto ja dokumentti saatavat näyttää aluksi täysin todenpitäviltä ja virheettömiltä. Voi olla jopa niin, että ison dokumenttikokonaisuuden tarkastus aiheuttaa väärän kuvan, jos dokumentin tarkastus on tehty pistokokein, tarkistamalla vain sadoista yksittäisistä tiedoista osa. Näin menettellen on mahdollisen virheen vaara suuri, sillä tämä yksikin tieto saattaa aiheuttaa mittavia ongelmia kerrannaisvaikutuksena. Onkin siis tarkkaan mietittävä, mihin luottaa. Tämä oli varsin selvästi havaittavissa työn kuluessa.

Työn alkuvaiheessa pyrin analysoimaan kaikkia käytössäni olleita dokumentteja ja tietolähteitä. Pohdin dokumenttien luonnetta ja sitä, mikä dokumentti vastaa mihinkin kysymykseen. Lisäksi punnitsin hyvinkin tarkkaan dokumenttien luotettavuutta. Työn alussa on tosin vaikeaa arvioida kunkin dokumentin hyödyllisyyttä, sillä käsillä olevan ongelman ja järjestelmän luonteesta ei vielä ole selkeää käsitystä. Tämä tosin on tarpeellista pohdintaa ja sitä voi ja tulee myöhemmin tietämyksen kasvaessa tarkentaa. Suppeasta käsityksestä huolimatta projektilla on yleensä selkeä suunta ja raamit, niin LSB:n tapauksessakin. (Demeyer Ym. 2008: 61.) Jätinkin osan dokumenteista tarkemman tarkastelun ulkopuolelle. Tällaisia dokumentteja olivat mm. rakennusautomaatioon liittyvät dokumentit, aseman rakennusteknilliset asiakirjat ja osa sähköpiirustuksista. Tietolähteinä olivat myös keskustelut tilaajan edustajien kanssa. Keskustelua olisin voinut käyttää myös työn alussa laajemmin hyväksi ja haastatella esimerkiksi käyttöhenkilöstöä. Katsoin kuitenkin jostain syystä, etten kyseisessä vaiheessa pystyisi löytämään haastattelulle sopivia kysymyksiä yms. Tämä todennäköisesti olisikin pitänyt paikkansa, ainakin jossain määrin. Lisäksi niiden keskusteluiden nojalla, joita työn alussa kävin VE:n henkilöiden kanssa, kuva LSB:n tilanteesta oli kohtalaisen synkkä. Käsitys kuitenkin tuntui osoittautuvan lopulta hieman turhan negatiiviseksi. Kyseisen kaltainen negatiivisen kuvan syntyminen ja antaminen on tosin täysin normaalia, ja tämä ilmiö tulisi-kin tiedostaa. (Demeyer ym. 2008: 75)

#### 4.4 Alkutietojen kerääminen ja analysointi

Työn alussa on syytä kerätä kaikki mahdollinen aineisto käyttöönsä ja ainakin silmäillä se läpi, jolloin saa käsitystä kohteen laadusta (Demeyer ym. 2008). LSB:n valvomohuoneen arkistosta löytyi yhteensä neljä mappia asemaa koskevia asiakirjoja sekä arviolta kaksi riisiä ilmeisesti vanhoja luonnoksia ja muutostyön aikaisia vedoksia, ”jätepapereita”, pahlilaatikossa. Otin kansiot käyttööni ja silmäilin aluksi läpi nämä asiakirjat. Näistä selvisi mm. että LSB:llä on käytössä rakennusautomaatiota, joka on muutaman tilatiedon osalta yhteydessä Simatic S5 -järjestelmään, josta nämä tiedot välitetään valvomoon mutta joita ei käytetä paikallisesti aseman hallinnassa. Lisäksi selvisi, että kaikki moottoriventtiileiden ohjaukset tapahtuvat releillä logiikan antamien ramppiaikapulssien mukaisesti, ja moottoriventtileiltä on takaisinkytkentä logiikalle (auki/kiinnitilat). Myöhemmin tarkistin myös edellä mainitun jätetierikasan, mutta siitä ei lopulta löytynyt mitään kovin mullistavaa, vain muutama projektiin osallistuneen henkilön nimi ja muutama päiväys pienille muutoksille.

Ensimmäisellä viikolla koostin myös dokumenteista aikajanaa aseman muutoksista ja historiasta (liite 2). Tämä tapahtui paljolti lähdemateriaalin läpikäymisen sivutuotteena, mutta antoi mielestäni syventävää tietoa aseman nykyisestä tilanteesta. Aseman historia alkaa vuodesta 1994, jolloin osa rakennus- yms. suunnitelmista on päivätty. Aseman käyttöönotto on tapahtunut vuonna 1995, ja ensimmäisessä kokoonpanosta on ilmeisesti puuttunut kaukokäyttö. Lisäksi aseman kautta voitiin siirtää lämpöä vain Helsingin Energian verkosta Vantaan Energian verkkoon, koska aseman alkuperäisestä prosessista puuttui kokonaan pumppu neljä. Aseman alkuperäisen ohjelman on ilmeisesti toimittanut Siemens, joskaan tästä ei voi olla täysin varma.

#### 4.5 Symbolitauluja vanhoista dokumenteista

Koska lähtöolettamuksena oli, että kaikki sähkökuvat pitävät paikkansa, keräsin sähkökuvista lähdöt ja tulot ja nimesin ne. Lisäksi LSB:n valvomohuoneen arkistosta löytämästäni alkuperäisestä lähdekoodista vuodelta 1995 keräsin vastaavat nimet ja tiedot ja missä näitä oli käytetty. Tiedot syötin taulukkolaskentaohjelmassa taulukkoon siten, että nämä vastasivat toisiaan. Myöhemmässä työn vaiheessa selvisi kuitenkin, etteivät sähkökuvatkaan kaikin osin pidä paikkaansa, sillä ohjainyksikön I/O:n johdotusten osal-

ta nk. ristiinkytkentätaulukot olivat vanhentuneita. Tämä johtuu aseman läpikäymistä uudistuksista ja päivityksistä.

Tutustuin ohjelman vanhoihin versioihin ja koostin näistä kohtalaisen hyvin kommentoiduista versioista käsin symbolitaulut merkkereille. Saatuaani käyttöön nykyisen käytössä olevan ohjelman oli minun kuitenkin todettava, etten voi luottaa kyseisiin symbolitauluihin, koska ohjelmien sisällössä oli valtava ero. Hylkäsin harkinnan jälkeen koostamani symbolitaulut senhetkisen tiedonpuutteen nojalla ja ohjelmaversioiden suuren eroavuuden takia sekä tästä mahdollisesti aiheutuvien virheiden pelossa.

Palasin vanhoista kommentteista koostamiini symbolitauluihin vielä työn lopussa selvitettyäni ohjelmakoodin ja koostettuani omat uudet symbolitaulut. Näitä vertasin joiltakin osin vanhoihin ja sainkin muutamien muuttujien ja muistipaikkojen kohdalle lisäselvennystä ja varmennusta omaan tulkintaani eri muuttujien luonteesta.

Tämä vanhojen symbolitaulujen koostaminen oli hyvä tehdä, sillä näin tuli ohjelma kertaalleen silmäiltyä läpi ja tällöin muodostui edes jonkinlainen näppituntuma ohjelmaan. Lisäksi työn myöhemmässä vaiheessa vanhoista kommentoiduista symbolilistoista on apua arvioitaessa nykyisen ohjelman tulkintojen oikeellisuutta.

#### 4.6 Nykyisen ohjelman arviointi

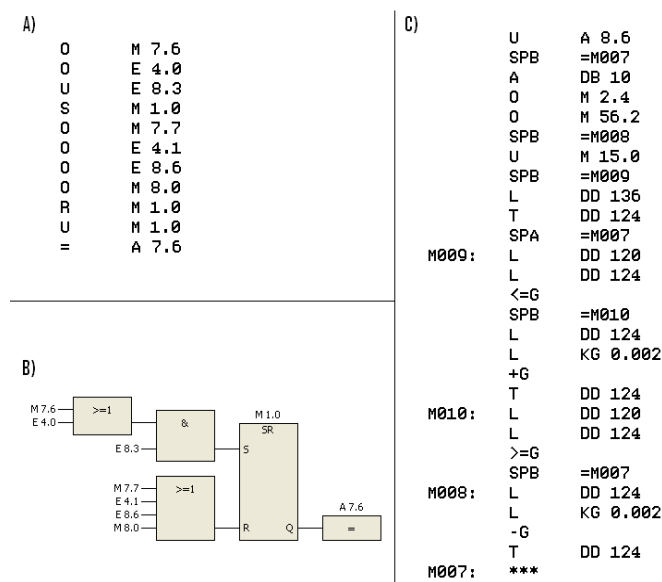
Heti kohta aloitettuani työtä kävimme lataamassa nykyisin käytössä olevan ohjelmaversion LSB:ltä. Edellisenä päivänä oli VE:n sähköisestä arkistosta löytynyt vielä yksi versio ohjelmasta vuodelta 2005, ja tämäkin oli kommentoimaton käyttöversio ilman symbolitauluja. Ladattuamme nykyisen ohjelman Siemensin Simatic S5 -logiikan muistista ohjelmointilaitteelle kokeilimme sen yhtenevyyttä vuoden 2005 version kanssa ja totesimme, että ohjelma oli muilta osin muuttumaton, paitsi että tiedostoyksiköiden data oli muuttunut, mikä oli odotettavissa, sillä ohjelma tallentaa mittaustulokset yms. tiedostoyksiköihin. Tästä menettelystä on tiettyjä etuja, mm. mittausarvon muutoksen estäminen ohjelmasyklin aikana. Tämän lisäksi vertasin paremmin ohjelmaa vielä S5/S7 for Windows -ohjelmalla, jossa ohjelma osaa kertoa myös, minkä tyyppisiä erot ovat koodissa, kts. luku 4.2.1. Tällä vertailulla selvisi, että ohjelma on tiedostoyksiköiden

lukuarvoja lukuun ottamatta pysynyt muuttumattomana verrattuna vuoden 2005 versioon.

Tosin tästä ei käytännössä ole hyötyä, koska vuoden 2005 versiokin on täysin kommentoimaton ja ilman symbolitauluja. Tiedon perusteella voidaan kohtuullisen turvallisesti olettaa, ettei asemalle ole tehty muutoksia vuoden 2005 jälkeen. Asema on käytönotettu vuonna 1995, ja viimeinen ohjelmamuutos on tehty 2005, eli asema on käynyt läpi kymmenen vuotta muutoksia ilmeisen huonolla suunnitelmallisuudella ja dokumentoinnilla.

#### 4.7 Ohjelman takaisinmallintamisesta

Jaoin ohjelman karkeasti kahteen osaan, joista toinen oli ohjelmayksiköiden (PB) sisältämä "perusohjelma", josta melko tyypillinen esimerkki on nähtävillä kahdessa Euroopassa yleisestä esitysasussa kuvan 11 A- ja B-kohdissa. Toinen on "edistynyt ohjelma" kirjoitettuna toimintayksiköihin (FB). Tarkoitin tällä ohjelman osia, joissa käytetään muitakin kuin loogisia funktioita ja ajastimia. Näistä korkeamman tason ohjelmarakenteita sisältävistä ohjelman osista koostuu noin 60 prosenttia ohjelman ohjelmariveistä, kun otetaan rivimäärässä huomioon myös käytetyt tiedostoyksiköt.



Kuva 11. Esimerkki ohjelman eroista. Kohdat A ja B ovat sisällöllisesti samat ja esitettyinä AWL- ja FBD-muodoissa. Kohdassa C on esitetty ote "edistyneestä ohjelmasta" AWL-muodossa.

Heti ohjelman tarkastelun alussa oli havaintona, että tiedostoyksikkö DB2 on jätetty käyttämättä ohjelmassa, kuten aikaisemmin tässä dokumentissa on mainittu. Tämä tiedostoyksikkö on Simatic S5-135U -proessoreissa varattu jonkinasteiseksi symbolitauluksi. Tähän ohjelmayksikköön en käyttänyt kuitenkaan enää enempää aikaa, kun tiesin, ettei kyseinen yksikkö ole käytössä. Sisällöllisestikin tämä yksikkö oli selvästi vanhentunut, koska se on ilmeisesti vuoden 1995 asussa. Samoin selvisi, ettei ohjelma käytä toimintayksikköä FB70. Tämän yksikön sisällöksi paljastui myöhemmin pumppu neljän (P4) ohjaus sellaisessa tapauksessa, jossa HE:n virtaama ei vapaalla kierrolla riitä, ts. säätöventtiili V14 avautuu täysin eikä täten pysty enää säätämään virtausta halutulle tasolle. Syy, miksi FB70 ei ole käytössä, ei ole tiedossa, joskin vanhoissa ohjelmaversioissa on maininta, että kyseessä oli varaus.

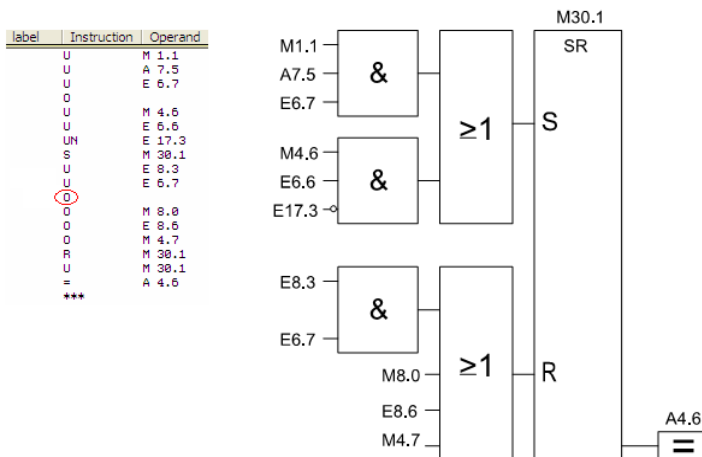
Suurimmiksi avaimiksi LSB:n takaisinmallinnukseen osoittautuivat iFIX-kaukokäyttöjärjestelmän tietokannasta tehty listaus, josta selvisivät tärkeimpien mittausten tiedostoyksikköosoitteet sekä muutamien keskeisten muuttujien nimet. Lisäksi listauksesta ilmenivät merkkerit, joiden tieto lähetettiin tai joiden tieto tuli kaukokäyttöjärjestelmästä. Lopulta uskoin vanhahkon I/O-listauksen pätevyuden tarkasteltuani kaikkia muita käytössä olevia dokumentteja, mikä helpotti osaltaan merkittävästi työn tekemistä. Edellä mainittujen tietolähteiden merkitys syntyi siitä, että näiden avulla sain hieman vinkkiä kunkin tuntemattoman merkkerin ja tuntemattoman muuttujan suhteesta tunnettuihin toimintoihin ja täten helpommin lokeroitua nämä oikeaan viitekehukseen.

#### 4.8 Alkuvaiheen harha-askeleita

Käytyäni aluksi ohjelmakoodin lävitse yritin ensimmäisessä vaiheessa mallintaa ohjelmakoodia suoraan tietokoneella vuorovaikutuskaavioiksi, mutta totesin tämän lähestymistavan vaativan enemmän ymmärrystä ohjelman rakenteesta ja toiminnoista, kuin minulla oli tuolla hetkellä käytettävissäni, sillä soveltuvan ohjelman puutteen vuoksi kaavioiden piirtäminen oli aikaa vievää, ja kun tietoon tuli jatkuvasti tarkennuksia, oli tämä liian raskas lähestymistapa. Päädyinkin ratkaisuun, jossa tulostin ohjelmakoodin paperille. Tämä osoittautui hyväksi ratkaisuksi, joskin paperille ohjelman avaaminen oli yhtäläillä työlästä, sillä ohjelmakoodia oli tulostettuna noin 65 sivua, johon ei sisälly tiedostoyksiköitä.

Kommentoin koodia siis paperille ohjelmakoodin viereen. Aluksi vain kommentoin yksittäisiä rivejä, sillä yhtenä haasteena työn tekemisessä oli hyvin vähäinen kokemukseni käskylistä muotoisesta ohjelmoinnista ja etenkin Simatic S5:n käskyistä. Opetuksessa käytettävänä laitteistona oli omassa yksikössäni ollut jo pitkään opetuskäytössä Siemensin logiikoista vain S7-tuoteperheen laitteita, ja näiden ohjelmointi tapahtuu osittain hieman eri komennoilla ja rakenteilla. Jouduinkin siis opiskelemaan Simatic S5:n ohjelmointia työn alussa.

Käänsin ohjelmaa soveltuvin osin FBD-muotoista esitystapaa mukailevaksi, sillä hyvin suuri osa ohjelmasta oli kirjoitettu AWL-muodossa, joka antaa ohjelman tekijälle vapauden tehdä koodia, mikä ei täytä FBD-muodon vaatimuksia ja esitystapaa, ja koodia jota ei voi ohjelmallisesti FBD-muotoon muuttaa. Käsin kuitenkin pystyin näin perusohjelman osalta tekemään, sillä ongelmana oli useampi tallennuskäsky yhdessä segmentissä tai muu muotovirhe koodissa, esimerkiksi ylimääräinen or-kärky (O) kuten kuvaan 12 on punaisella ympyröity. Tässä kuvassa on esitetty vasemmalla yhden moottorinohjaussegmentin koodi AWL-muodossa ja oikealla esitettyä tätä vastaava tulkintani FBD-muodossa.



Kuva 12. Esimerkki koodista AWL- ja FBD-muodoissa sekä ohjelmallisen muunnoksen estävästä muotovirheestä.

Päädyin tähän esitystapaan lopulta Linnan ehdotuksesta. Perusohjelman osalta tämä oli helppo ja hyvä tapa toimia, sillä usein FBD-muotoisen ohjelman lukeminen ainakin tähän tai muuten digitaalisen elektroniikan piirikaavioihin perehtyneelle on helppoa ja usein jopa huomattavasti havainnollisempaa kuin sanallinen selvitys.

Perusohjelman piirikaavioesityksestä laadin kirjalliseen selvitykseen perusohjelman osalta luettelomerkein jäsennellyksi listaksi erilaisista osaohjelmien loogisista vaatimuksista tietyissä tilanteissa. Tästä on esimerkki esitetty kuvassa 13.

***Pumput 1-3 automaattiohjaukselle PB2/3-5***

- Yksi tapa
- Lukitukset on ilmeisesti rakennettu väärin, koska näyttäisi että ei ole mahdollista ajaa edes paikallisesti pumppuja automaattiajolla jos KK on estetty.
- Tila AUTO:lle
  - E17.3 KK estetty käskyn on oltava passiivinen
  - Lisäksi toinen seuraavista on oltava aktiivisena
    - Px Auto käsky
    - KK ohjaus Px Auto

Kuva 13. Erään logiikkaohjelman toiminnon loogiset vaatimukset.

Ongelmana oli, että osaan ohjelmanosista tämä esitystapa antoi selkeän kuvan, mutta osaan se näin jälkikäteen tarkasteltuna ei toimi niin hyvin kuin olisin toivonut, kuten kuvassa 14 on havaittavissa.

***Avaa Vantaan menosulkuventtiili V12 A7.0 PB1***

- Venttiilin tila on käynnistys ehtona pumpuille 1 – 3
- Käynnistämiseen on kaksi tapaa. Toinen tavoista on automaattiohjaus ja toinen on käsinohjaus logiikan kulloisenkin tilan mukaisesti.
- Tapa 1 – Auto
  - Logiikan tulee olla Automaattiohjaus tilassa A7.5
  - ja logiikassa on Autostart osoitus päällä A7.6
  - Lisäksi on oltava valittuna toinen seuraavista
    - Ajosuunta Vantaalta Helsinkiin A8.7
    - Ajosuunta Helsingistä Vantaalle A8.6
- Tapa 2 – Käsini (kaukokäytön kautta)
  - Logiikka on oltava käsin tilassa eli osoitus A7.4 on tällöin aktiivinen
  - Avaa Vantaan menosulkuventtiili V12 KK ohjaus M7.0 aktiivinen

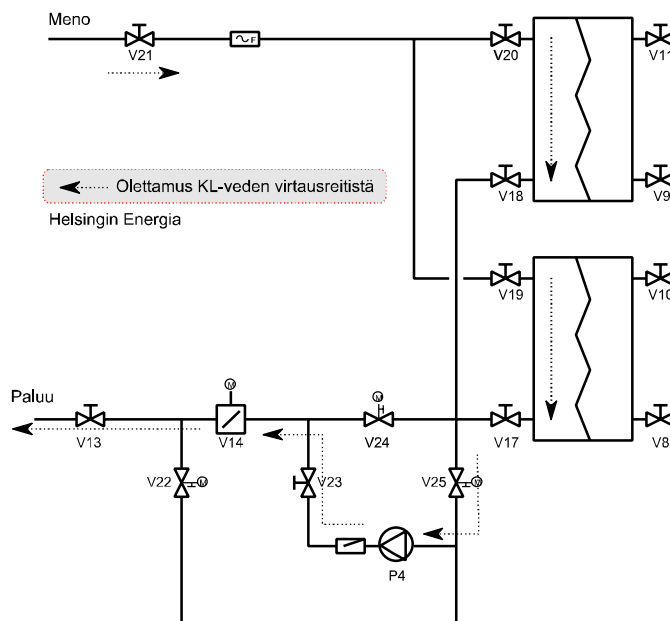
Kuva 14. Erään logiikkaohjelman toiminnon loogiset vaatimukset.

#### 4.9 PI-kaavio

Dokumenttejä tutkiessani löysin vanhan PI-kaavion prosessista sekä tietenkin valvomossa käytettävän ylimalkaisen prosessikuvan. Koska oli tiedossa, että asemalle on tehty useita muutoksia eikä löytynyttä PI-kaaviota ollut päivitetty, katsoin tarpeelliseksi sen tarkastamisen. Sama PI-kaavio löytyi MGF-muotoisena tiedostona PIKAAVIO.DOC, jonka avaaminen on selostettu luvussa 4.2.2. PI-kaavion todenpitävyyden varmistamisen aloitin valokuvaamalla aseman prosessin järjestelmällisesti, lisäksi otin valokuvia toimilaitteisiin liittyvistä tunnisteista ja tyyppikilvistä. Osan jouduin luonnostelevaan paperille käsin paikan päällä, koska valokuvaaminen ei tilasta johtuen ollut mahdollista. Lopuksi omalla työpisteelläni selasin valokuvia ja "kävelin" prosessia läpi useampaan kertaan, samalla vertasin nykyisiä mittauksia ja putkistoja vanhaan PI-kaavioon. Varsin pian kävi ilmi, ettei vanha PI-kaavio pitänyt kaikin osin paikkaansa, sillä osa mittauksista oli piirretty vanhassa kaaviossa väärään paikkaan. Lisäksi muutama paineekytin ja kaikki osoittavat mittarit oli jätetty piirtämättä vanhaan kaavioon. Tästä syystä jouduin piirtämään myös PI-kaavion uusiksi, minkä toteutin edellä mainittuja valokuvia hyväksi käyttäen. Piirtämäni uuteen PI-kaavioon, joka on liitteenä 1, lisäsin myös nk. paineenpitopiirin varsinaisen lämmönsiirrinprosessin rinnalle. Kaukolämpöveden virtaamisen suunnat oli merkitty putkien ulkopintaan, ja aluksi piirsinkin kaavioon suuntanuolet näiden mukaisesti, sillä ymmärsin prosessin rakenteen väärin työn alussa käytyjen keskusteluiden ja jo mainittujen virtausmerkintöjen takia.

Ihmetystä herätti monimutkainen moottoriventtiileiden ja putkien verkko pumpun neljä (P4) ympärillä. Ihmettelin ennen muuta, miksi taajuusmuuttajaohjatun P4:n tuottamaa virtausta rajoitetaan kuristamalla säätöventtiilillä V14 ja mikä tehtävä venttiilillä 22 on prosessissa. Virheellinen olettamani KL-virtauksen reitistä oli kuvan 15 kaltainen.

Myöhemmin selvisi, ettei kuvan 15 mukainen toiminta ole käytössä ja että prosessi on piirretty P4:n ympärille oikein. Olin myös ohjelmakoodin perusteella ruvennut epäilemään, että jotenkin tämä tulkintani aseman normaalista toiminnasta ei ole järkevä, ja kun otin asian esille, selvisi, että kyseessä on jonkinlainen tehostepumpputoiminto, joka ei normaalisti ole käytössä. Tämäkin tieto edellä mainitusta funktiosta ja toiminnasta varmistui ja tarkentui oikeaksi myöhemmin. Oikeat KL-virtausten reitit ovat nähtävissä liitteissä 3 ja 4. Näihin on PI-kaavioon merkitty samankaltaisesti kuin kuvassa 15 virtauksen reitti riippuen kaukolämpöä luovuttavasta osapuolesta.



Kuva 15. PI-kaaviota tehdessä ollut väärä käsitys KL-virtauksen reitistä.

PI-kaaviosta oli luonnollisesti suurta hyötyä ohjelman ja etenkin aseman toiminnan selventämisessä, samalla PI-kaavio palvelee tulevaa aseman ohjauslogiikan vaihtoa ja mahdollisia toiminnan päivityksiä. Piirtämäni PI-kaavio on liitteenä 1.

#### 4.10 Löytyneiden tietojen tallentaminen

Yksi tärkeä asia työtä tehdessä oli mahdollisimman hyvien työnaikaisten muistiinpanojen tekeminen. Nämä käsittivät kaikissa laadituissa dokumenteissa aina tiedon siitä, mihin lähteeseen kukin dokumentti tai yksittäinen tieto perustui. Tämä on tärkeää ennen muuta projektin alussa, kun rakennetaan pohjaa laajemmalle ymmärrykselle kohteen toiminnasta. Tärkeätä on käännteissuunnittelun onnistumisen kannalta kirjata

ylös myös kunkin tiedon ja väittämän lähde, ei niinkään asiakkaan vakuuttamiseksi tiedon todenperäisyydestä, vaan pelkästään oman työn helpottamiseksi, sillä monesti työn aikana syntyneet tiedot, olettamukset ja väittämät tarkentuvat, ja ajoittain aina jonkin tiedon oleellinen tarkentuminen johtaa jonkin muun olettamuksen vääräksi toteamiseen. Tällöin on ehdottoman tärkeää, että on tieto, mistä kohden tieto on peräisin ja mihin tämä väittäminen on perustunut. Muuten saattaa käydä, ettei aikaisemman idean ja tiedon alkuperää enää muistakaan, ja tällöin muiden muuttuneeseen tietoon perustuvien väittämien korjaaminen on hankalaa ja pelkästään tekijän muistin varassa.

Tähän tiedon tallentamiseen olisikin omiaan jokin tarpeeksi löyhästi määritelty tietokantasovellus, sillä tämänkaltaisen ristiin menevän taulukon ja asiakirjakokonaisuuden ylläpito on vähintäänkin työlästä ja useimmiten turhauttavaa, sillä aikaa menee turhan paljon vain tiedon päivittämiseen, ei uuden tiedon tuottamiseen. Tämä turhautuminen johtaa myös helposti pohdinnan ja työolettamusten laatimisen unohtamiseen, mikä lopulta näyttelee yllättävänkin suurta osaa tämänkaltaisessa takaisinmallintamisessa. Lisäksi on hyvä pitää kirjaa kysymyksistä, joita muodostuu työtä tehdessä.

Itse havaitsin tämän puutteen työn puolesta välissä, kun alkumatkasta olin luottanut vain omaan muistiini. Kun otin kysymysten kirjaamisen käyttöön, kirjoittelin niitä aluksi lähinnä papereiden marginaaleihin. Tämäkin tapa on tyhjää parempi ja varmasti toimii muutamien sivujen dokumenteissa, mutta kun papereita on kymmeniä, tämänkaltaisten muistiinpanojen tekeminen ja etenkin käyttö on turhaa, sillä ne häviävät epäjärjestyksensä takia tiedon paljouteen. Väitän tässä työssä saamieni kokemusten pohjalta, myös että työnaikaisten kysymysten laadintaan perinteinen sivukooltaan tarpeeksi suuri ruutuvihko on tietokonetta parempi muistiinpanojen kirjaamiseen, ellei käytössä ole sovellusta, johon koko projektin voi kirjata ilman, että tarvitsee hyppiä ohjelmasta toiseen. Lisäksi ohjelman on oltava äärimmäisen joustava tai suunniteltu nimenomaisesti automaation takaisinmallintamis- ja uudelleensuunnittelutyöhön.

Paperin ja kynän etuna on sen monipuolisuus, käyttäjä ei ole sidottu tiettyihin ennalta määrättyihin toimintoihin ja tapoihin kirjata asioita. Käyttäjällä voi helposti kirjoituksen lisäksi luonnostella kaavioita, vetää nuolia vaikka sivulta toiselle jne. Haittana tässä työskentelytavassa on dokumenttien hallinnointi ja päivittäminen. Kuten edellä asiaa jo käsiteltiin, tulee tästä suunnattomasti aikaavievää ylimääräistä työtä, jos verrataan

vastaavalla vapaudella toteutettuun tietokoneohjelmaan, jossa on tietokanta-ominaisuuksia.

#### 4.11 Logiikkaohjelmaa takaisinmallintaessa tehtyjä huomioita

Alkutietojen nojalla oli odotettavissa, että logiikkaohjelmasta löytyy paljon virheitä ja päällekkäisyyksiä, osin tämä olettaus pitikin paikkansa. Logiikkaohjelmaa tulkitessa oli jo varhaisessa vaiheessa nähtävillä, että aseman ohjelmaa oli moneen otteeseen muokattu ja monen eri henkilön toimesta. Parhaiten tämä näkyi mm. pienistä ohjelmarakenteiden eroista, joista oli selvästi huomattavissa, kuinka toisaalla oli saman asian tekemiseen käytetty useampi rivi AWL-koodia ja toisaalla oli pärjätty puolta vähemmällä rivimäärällä. Lisäksi muutamia suoranaisia ohjelmointivirheitäkin löytyi.

Ohjelmassa oli myös paljon toistoa, josta selvästi huomasi, että ohjelmaa oli rakennettu leikkaa ja liimaa –periaatteella. Tällaista ohjelmointia pyritään yleisesti välttämään, koska se aiheuttaa ohjelmaan turhaa pituutta ja sekavuutta, mikä johtaa toisaalta herkästi tulkinnaivaikeuteen ja ylläpidon resurssien kasvamiseen. Funktiokutsuja käyttäen ohjelman rakenne olisi ollut jäsennellympi ja mahdollisia muutoksia tehdessä olisi muutoksien tekeminen vain yhteen paikkaan helpompaa kuin muutosten korjaaminen useaan paikkaan. Tosin on huomioitava, että myös funktiokutsun yhteydessä sijaitsevat mahdolliset lähtöarvot ja muut sellaiset parametrit saattaa olla tarpeen muuttaa, mutta silti funktioiden käyttö olisi usein selkeämpää. (Demeyer ym. 2008)

##### 4.11.1 Tehonrajoitus virtauksen mukaan

Ohjelmaa analysoitaessa kävi ilmeiseksi, että kaukolämpövirtaukseen perustuvan tehonrajoituksen tarpeen havaitseva ohjelman osa oli rakennettu väärin. Tämä kokonaisuus on rakennettu neljästä irrallisesta ohjelman osasta, jotka on liitetty toisiinsa boolean-arvon saavalla merkkerillä. Merkkerit on kuitenkin liitetty keskenään ristiin, jolloin tarkoitettu toiminta ei toteudu virtauserosta huolehtivan venttiilin ollessa lähes täysin auki.

Lisäksi virtauseron tarkistus prosessista mitatuista arvoista on toteutettu kertomalla virtauksien erotus luvulla 15 ja tarkistamalla, onko tulos pienempi kuin ajosuunnan

mukainen virtausarvo. Tämä ohjelmarakenne virtauseron prosentuaalisen arvon havaitsemiseksi on ohjelmaa huoltavalle henkilölle hankala, ja haluankin kyseenalaistaa rakenteen käytännöllisyyden, kun vastaavaan toiminta saavutettaisiin selkeämmin hie-man toisella ohjelman rakenteella ja käyttäen kerrointa 1,0666.

Samana ohjelmaosuuden toisessa rakenteeltaan identtiseksi tarkoitettussa osassa, joka huolehtii tehonrajoituksesta virtauksien eron mukaan, kun aseman ajosuunta on päinvastainen, on myös selkeä ohjelmointivirhe. Tässä kaksiosaisesta ohjelmaosasta ensimmäisen osan aloittava ajosuunnan tarkistus on ohjelmassa ilmeisesti vahingossa merkitty hyppäämään yhden rivin liian ylös ohjelmassa. Tämä virhe aiheuttaa tehonrajoituksen aina VE:n säätöventtiilin ollessa yli 94 % auki. Tämän luvun ensimmäisessä kappaleessa esitetty virtauseron prosentuaalinen erotus on myös tässä ohjelmoitu väärin.

Ohjelmaa takaisinmallintaessa samalla saatiin tieto, ettei aseman virtauksen tehonrajoitus olekaan 5 % vaan 6,66 %.

Lisäksi aseman tehoinmittauksissa on ohjelmaa tehtäessä käytetty paljon toistoa, vaikka ohjelmanrakenne ja suoritettava toimenpide olisi toteutettavissa yhdellä kutsuttavalla funktiolla, jolloin vain raja-arvot ja halutut muistipaikat asetetaan kutsuttaessa.

#### 4.11.2 Virtauksen säätö

Lämpöä luovuttavan kaukolämpöverkon virtauksen säätö on riippuvainen lämpöä vastaanottavan kaukolämpöverkon pumppujen verkkoon tuottamasta virtaamasta. Lämpöä luovuttavan kaukolämpöverkon virtauksen säätö tapahtuu releohjatulla säätöventtiilillä, jolloin releohjelmalle annetaan logiikasta ohjauspulssi, jonka pituus määrittää lopulta venttiilin asennon. Tämä virtauksen säätö on rakennettu ohjelmassa PID:n kaltaisella säädöllä ja ajastimella, joka antaa ehtojen täytyessä ramppiaikapulssin venttiilin releohjaukselle. Säädön tarvitsema ohjelman osuus on rakennettu useampaan toimintayksikköön (FB) muiden ohjelmanosien joukkoon, mikä lisää vaikeutta tulkita ohjelmaa esimerkiksi virhetilanteissa.

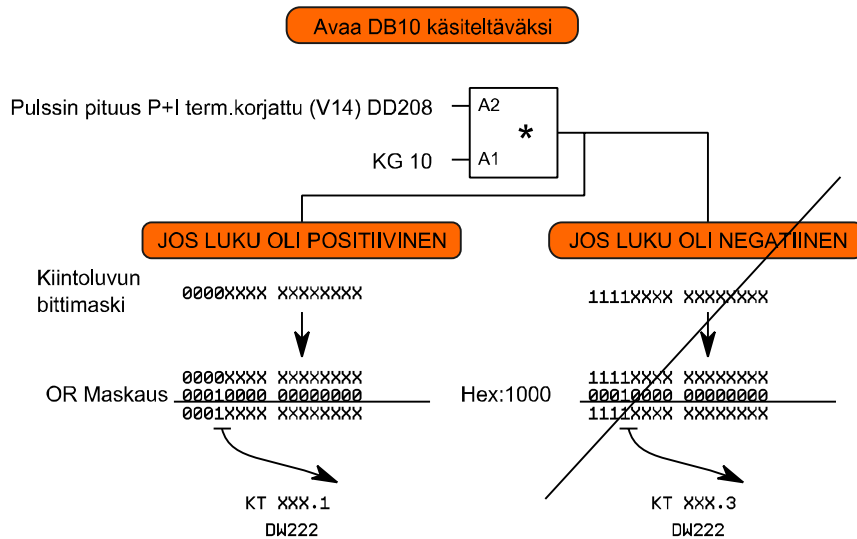
#### 4.11.3 Säättäjän ohjausarvon muuntaminen aika-arvoksi

Lisäksi ohjelman osa, joka huolehtii säättäjän liukulukumuotoisen ohjausarvon muuntamisesta Simatic S5 -logiikan käyttämäksi aika-aelimen aika-arvoksi aikakertoimiseen, on rakennettu sekavasti ja virheellisen aika-arvon mahdollistavalla tavalla (kuva 16). Näin on siis tapauksessa, jossa ohjausarvon etumerkin käsittelystä (venttiili auki tai kiinni) huolehtiva boolean-arvon saava merkki jostain syystä saisikin ohjausarvosta poikkeavaa etumerkkiä osoittavan arvon tms. Tällöin aika-aelimen (ajastimen) aika-arvon aikakertoimeksi muodostuu luku 3, joka tässä tapauksessa tarkoittaa 100-kertaista aikaa verrattuna oikeaan aikakertoimeen, jonka tulisi olla luku 1 eli 100 ms. Tämä ohjelman osa yhdessä tapauksessa neljästä on esitetty kuvassa 16 korkeahkolla abstraktiolla. Kuvassa on nähtävissä or-maskaus heksalukuarvolla 1000 ja sen vaikutus aikakertoimeen.

Mahdollinen ongelma olisi poistettavissa, kun käsiteltävät alkuarvon aikakerrointa merkitsevät bitit nollattaisiin and-maskauksella eli otettaisiin luvusta itseisarvo ja vasta tämän jälkeen käytettäisiin or-maskauksella halutun aika-arvokertoimen asettamiseksi. Tällöin olisi ohjelma vähemmän herkkä mahdollisille virheille, mikäli ohjelmaa muilta osin muutettaisiin ja myöhemmin tehtävä boolean-arvoon perustuva tarkistus jäisikin pois tai muuttuisi nykyisestä. Mikäli aseman uudeksi logiikaksi valitaan Siemensin S7-tuoteperheen logiikka, on hyvä kiinnittää huomiota tähän muunnokseen, mikäli ohjelmaa ei kirjoiteta kokonaan uusiksi. Muunnos olisi mahdollista rakentaa myös erikseen kutsuttavaksi ja kutsuvaiheessa arvoiltaan aseteltavaksi funktioksi.

Hieman samankaltainen rakenne liukuluvun muuttamiseksi aika-arvoksi ramppiaikapulssia varten on myös kaikissa kaukokäytöstä tulevissa ramppiaikaan liittyvissä ohjauksissa.

V14 KIINNI OHJAUSARVON MUUNTAMINEN KT-MUOTOON  
 LSB S5 7.2011 RAMPPIAJAKSI. FB 20 seg. 8



Kuva 16. Suurella abstraktiolla esitetty esimerkki venttiilin säätäjältä tulevan liukulukumuotoisen ohjausarvon muuntamiseksi Simatic S5 -ajastimen ymmärtämään muotoon.

#### 4.12 Takaisinmallintamisen avulla tuotettu uusi dokumentaatio

Takaisinmallintamisen tuloksena aseman toiminnasta syntyi noin 100-sivuinen dokumentti, joka koostui pääosin erilaisista ohjelman toimintaa selventävistä kaavioista, taulukoista ja näihin liittyvistä huomioista ja ohjeista. Lisäksi dokumentin alkuun sisältyi toimintaselostus aseman perustoimintojen käyttämisestä.

Edellä mainitun dokumentin koostaminen alkoi vaiheittain, kuten koko takaisinmallinnus. Aluksi käsitys aseman logiikkaohjelman toiminnasta oli puutteellinen, varmaa tietoa ei ollut ollenkaan. Ohjelma itsessään oli lähes puhdasta AWL-muotoista ohjelmointia ilman kommentteja, kuten liitteen 5 sivulla 1 on nähtävillä. Ohjelman toimintaa ja rakennetta vaiheittain kommentoimalla ja tarkentamalla, kuten liitteen 5 sivulla 2 on nähtävillä, saavutettiin vaiheittain ymmärrystä ohjelman rakenteiden merkityksestä ja eri muistipaikkojen tehtävistä. Pelkkä kommentoitu AWL-muotoinen ohjelma ei vielä itsessään ole monimutkaisissa rakenteissa kovinkaan selvälukuista, kun kyseessä ei ole lukijan itse kirjoittama ohjelma. Tätä ongelmaa varten ratkaisuksi valikoitui liitteen 5 sivulla 3 esitetty sovellettu FBD-esitys, jolla ohjelma on jo visuaalisesti havainnollisempi. Tätä esitystapaa käyttäen myös monet pienistä koodinpaloista koostuvat laajemmat kokonaisuudet rupesivat selkeytymään, ja täten myös muistipaikkojen merkitys kirkas-

tui. Kun logiikkaohjelman eri toiminnot rupesivat erottumaan, aluksi varsin utuisina, oli seuraavana vaiheena kunkin osan kommentoiminen sanallisesti sovelletun FBD-esityksen alle, kuten liitteen 5 sivun 3 alaosassa on nähtävillä.

Näistä lopulta pystyi rakentamaan vielä hieman suurempia toimintokokonaisuuksia, joiden toimintaa selostettiin sanallisesti. Selostus tapahtui dokumentissa olevien kappa-leiden alussa. Lisäksi selvitystä ja toimintaselostusta varten tehtiin vuokaavioesityksiä aseman keskeisimmistä toiminnoista. Eräs tällainen keskeinen toiminto on nk. autotop, joka aseman ollessa automaattiohjauksella ajaa sen seis-tilaan. Tämän autostop-toiminnon vuokaavioesitys on nähtävissä liitteenä 6. Vuokaavioesitysten lisäksi doku-mentin ensimmäiset 20 sivua pitivät sisällään aseman varsinaisen toimintaselostuksen sanallisena esityksenä. Selostuksessa selvitettiin aseman perustoimintoja ja niiden vaa-timia oikeita toimenpiteitä, joista vain osa oli tiedossa työtä aloitettaessa. Näiden lisäksi selostuksessa oli esitettyä muun muassa logiikkaohjelmasta tulkittuja paineiden raja-arvoja, tietyillä ohjaustoimintojen yhdistelmillä toimimattomia toimintoja yms.

Työn tilaajalle luovutetun dokumentin loput lähes 80 sivua olivat liitteen 5 sivun 3 kal-taista tietoa ohjelman eri osien tarkemmasta toiminnasta tulevaa migraatiota helpot-tamaan.

## 5 Yhteenveto

### 5.1 Yleistä

Asemaa on nyt ajettu useamman kerran insinööriyössä laaditun dokumentaation luo-vuttamisen jälkeen ja on todettu, ettei aseman toiminnassa ole välitöntä vikaa. Työn alussa esille tulleet ongelmat johtuivatkin ilmeisesti aseman toimintojen syvällisemmän ymmärryksen puutteesta eivätkä niinkään ohjelmassa olevista virheistä.

Takaisinmallinnuksella syntynyttä uutta dokumentaatiota lämmönsiirraseman toimin-nasta pidettiin tilaajan palautteen perusteella varsin onnistuneena ja erittäin hyödylli-senä aseman käyttöä ja kehittämistä ajatellen.

Tulevina vuosina työn tilaajalla on kuitenkin edessään päätös aseman Simatic S5 -järjestelmän korvaamisesta uudemmalla järjestelmällä.

Yleisellä tasolla takaisinmallintamisen ja uudelleensuunnittelun rooli ja merkitys automaatiiossakin tulee korostumaan tulevina vuosina, kun automaationkin sovellukset laajenevat ja monimutkaistuvat sekä integroitujen MES- ja ERP-järjestelmien yleistyessä ja vanhetessa.

## 5.2 Ajatuksia migraatiota varten

Logiikan päivityksen tapahtuessa takaisinmallinnuksen jälkeen nk. black box -periaatteella muuttamatta logiikan ulkoista toimintaa saadaan migraatio suoritettua lyhyellä tuotantokatkoksella ja pienemmin resurssein. Ennen migraation toteuttamista on myös perehdyttävä aseman uudelleensuunnitteluun takaisinmallintamisella saatujen tietojen pohjalta, jotta migraatiossa ei tehdä mahdollisesti tarpeellisia uudistuksia estäviä tai vaikeuttavia ratkaisuja. LSB:n tapauksessa ollaan tällöinkin kuitenkin edelleen tilanteessa, jossa dokumentaatio aseman kytkennöistä on osin puutteellinen automaation osalta. Nämä dokumentaatiot olisikin päivitettävä pikimmiten migraation jälkeen, optimaalisessa tilanteessa tarkistettava jo ennen migraatiota, ymmärryksen varmistamiseksi.

Osa alkuperäisistä LSB:n toimilaitteiden paikallisohjauksista ohjaustaululla on ilmeisestikin johdotettu ohittamaan Simatic S5 -logiikka, lisäksi venttiileissä on käsipyörä käsin tapahtuvaa mekaanista käyttöä varten.

Uuteen logiikkaan ohjelmaa muunnettaessa on hyvä kiinnittää huomiota seuraaviin asioihin. Aseman ohjauksen ja säädön merkittävimmät osat on kirjoitettu 100 ms:n välein kutsuttaviksi, ja niissä monissa on käytetty ajastimina tapauskohtaisesti rakennettuja ohjelmarakenteita, joiden oikeaan muuntamiseen on kiinnitettävä erityistä huomiota. Lisäksi säätöalgoritmi on suurella todennäköisyydellä riippuvainen ohjelman kiertoajasta, kun puhutaan säädön optimaalisuudesta ja/tai robustisuudesta. Mikäli nykyisin käytettyä ohjelmarakennetta hyödynnetään, on varmistuttava tässäkin tapauksessa, että näytteenottotaajuus PI(D)-säätimelle pysyy samana tai ainakin on säädön toiminta testattava kunnolla ennen käyttöönottoa. Säädölle on tehty aseman alkuvai-

heen historiassa ilmeisesti paljonkin parametroidin muutoksia, jotta on saatu haluttu toiminta aikaiseksi. Lisäksi on kiinnitettävä huomiota ohjausarvojen muotomuunnoksiin liukuluvuista ajastimien aika-arvoiksi.

Kaukolämmön tehonmittaukset suorittavien ohjelmanosien muutoksessa on kiinnitettävä huomiota jo edellä mainittuihin tapauskohtaisesti rakennettuihin ajastimiin.

### 5.3 Takaisinmallintamisen vahvuuksia

Takaisinmallinnuksen avulla saadaan väritymätön kuva logiikan toiminnoista ja siihen kytketyn prosessin ohjauksesta.

Mikäli tarkasteltava ohjelma on halutun kaltainen, saadaan takaisinmallinnuksella ohjelman dokumentaation lisäksi käyttöön joukko näkemyksiä mahdollisista parannuksista.

Mikäli ohjattava prosessia ei voida keskeyttää tai testaaminen normaaliolosuhteissa ei anna tarvittavaa ymmärrystä ohjelmasta, saadaan takaisinmallintamisella automaatio-prosessista väritymätön kuva ainakin säätävän tai ohjaavan ohjelman osalta.

Mikäli tarkasteltava järjestelmä on pitkään jatkuneen virittelyn tulos johtuen prosessin luonteesta, saataneen takaisinmallinnuksella ymmärrystä myös prosessin toiminnasta ohjelman toiminnan rinnalle.

Edellä mainitut seikat helpottavat onnistuneen ohjelmistoalustan päivittämisessä.

Onnistuneesti toteutettuna takaisinmallinnus vapauttaa tietystä ohjelma-alustasta, koska ohjelman toiminnot selvitetään. Lisäksi prosessi antaa uudelleensuunnittelun myötä mahdollisuuden kehittää tarkastelun kohteena olevaa järjestelmää.

Takaisinmallinnus tukeutuu ainoaan täysin luotettavaan dokumentaatioon logiikkaohjelman toiminnasta.

#### 5.4 Takaisinmallintamisen heikkouksia

Mikäli suora ohjelmallinen käänös alustalta toiselle on mahdollinen ja sillä saavutetaan hyväksyttävällä tasolla toimiva ratkaisu ilman varsinaista ymmärrystä ohjelman toiminnasta, on takaisinmallintamiseen uhrattu aika jäänyt tappioksi.

Takaisinmallintaminen on varsin aikaavievää, ja tulokset ovat ennen työn aloittamista vaikeasti arvioitavia.

Automaatiossa käytettäville ohjelmistoalustoille käyviä ja takaisinmallintamista helpottavia ohjelmistoja on vaikea löytää. Universaaleja ohjelmia tiedonhallintaan projektia tehtäessä ei tiettävästi ole markkinoilla.

## Lähteet

Allworth S.T. & Zobel R.N.. 1991. Introduction to Real-time Software Design. 2<sup>nd</sup> edition. London: MacMillan education ltd.

Demeyer Serge, Ducasse Stéphane ja Nierstrasz Oscar. 2008. Object-Oriented Reengineering Patterns. Sveitsi: Square Bracket Associates.  
<<http://www.iam.unibe.ch/~scg/OORP/>> Ladattu 5.5.2011

John, Karl-Heinz & Tiegelkamp, Michael. 2001. IEC 61131-3: Programming industrial automation systems. Berlin: Springer.

Kaukolämmön käsikirja. 2006. Energiateollisuus Ry. Libris Oy, Helsinki.

Linna, Timo. 2011. Kunnossapitoinsinööri. Vantaan Energia Oy, Vantaa. Keskusteluita touko–marraskuu 2011.

Motus, Leo & Rodd, Michael G.. 1994. Timing analysis of real-time software. Pergamon. Trowbridge: Redwood Books. ISBN: 0080420265

Brisky, Russ. 2007. Simatic S5 to S7 migration, Siemens Energy & Automation Inc. Sähköinen dokumentti.  
<[http://www.automation.siemens.com/automation/jp/Documents/5\\_SIMATIC\\_S5\\_S7\\_renewal\\_manual\\_en.pdf](http://www.automation.siemens.com/automation/jp/Documents/5_SIMATIC_S5_S7_renewal_manual_en.pdf)>. ladattu: 28.10.2011

Siemens Oy. 2011a. Why to migrate?  
<[https://support.automation.siemens.com/WW/llisapi.dll/csfetch/27229299/migrieren\\_en.pdf?func=cslib.csFetch&nodeid=27239403](https://support.automation.siemens.com/WW/llisapi.dll/csfetch/27229299/migrieren_en.pdf?func=cslib.csFetch&nodeid=27239403)>. Luettu 11.8.2011

Siemens Oy. 2011b. Simatic S5.  
<[http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat\\_logiikat\\_simatic/s5.htm](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic/s5.htm)>. Luettu 11.8.2011

Siemens Oy. 1989. Simatic S5. Peruskurssi. Siemens Automaatiokoulutus. Kurssimateriaali. 2/1989

Strömman, Mika. 2002. Ohjelmoitavan logiikan ohjelmointi ohjelmistotuotantoprosessina. Espoo: Teknillinen korkeakoulu. Automaation tietotekniikan laboratorio.  
<<http://www.automationit.hut.fi/julkaisut/documents/PLC-ohjelmointi-MStromman.pdf>> Ladattu 9.5.2011.

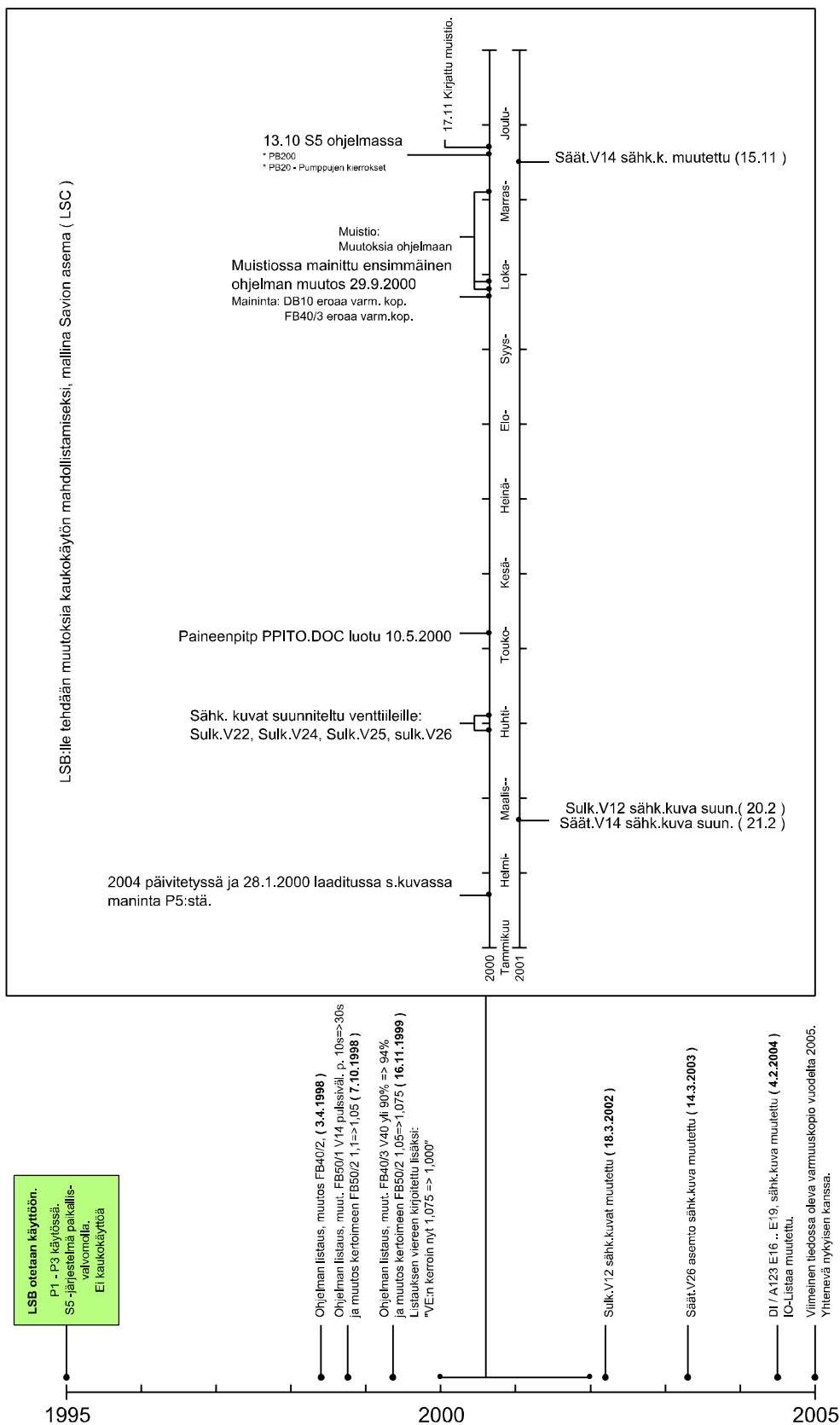
Vantaan Energia Oy. 2011. Yhteiskuntavastuu.  
<<http://yhteiskuntavastuu2010.vantaanenergia.fi/Sivut/default.aspx>> Luettu 4.5.2011

Wikipedia 2001a = Lehman's laws of software evolution. 2011. Wikipedia.  
<[http://en.wikipedia.org/wiki/Lehman%27s\\_laws\\_of\\_software\\_evolution](http://en.wikipedia.org/wiki/Lehman%27s_laws_of_software_evolution)> Luettu 8.8.2011.

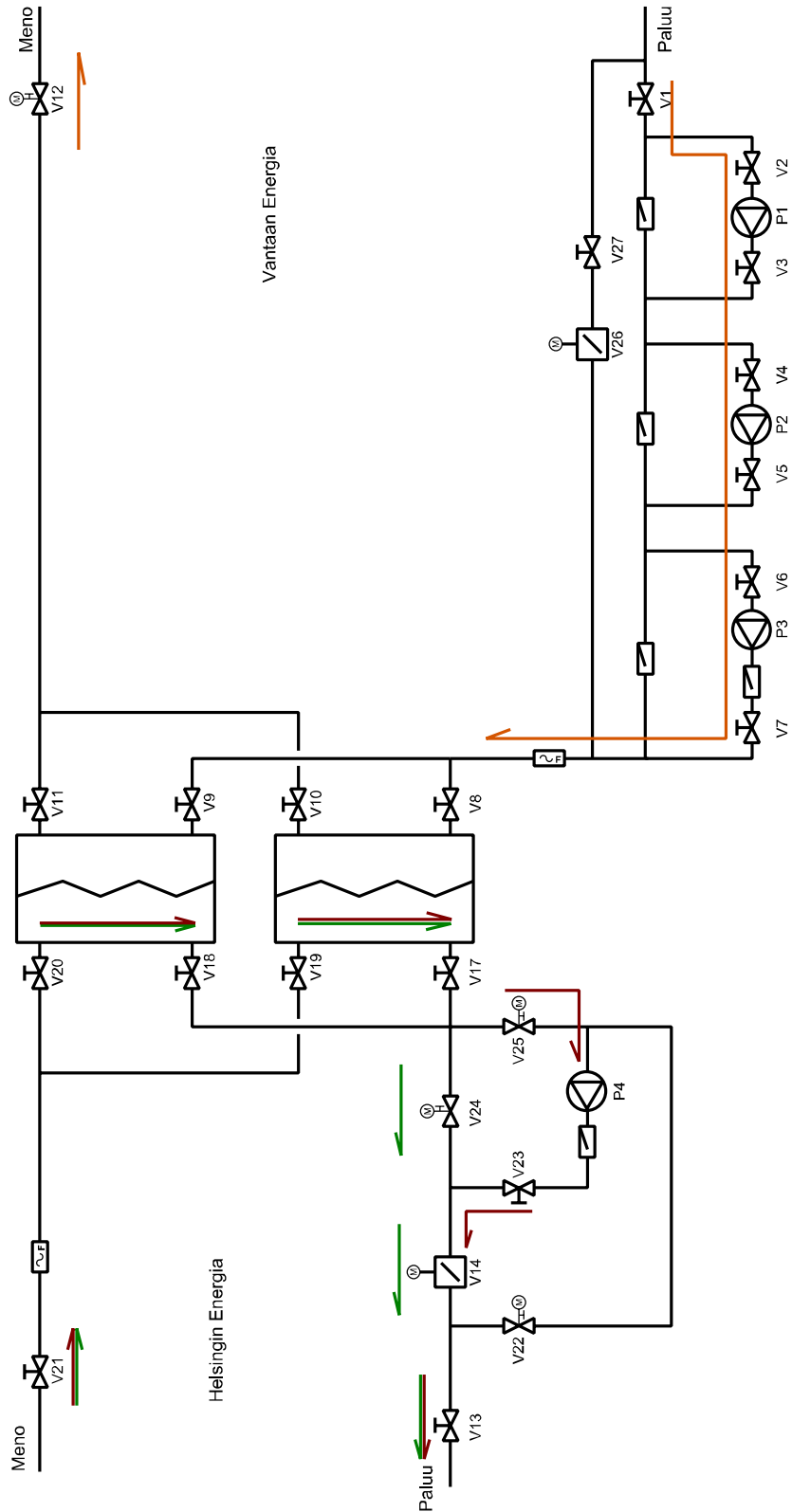
Wikipedia 2001b = Reverse engineering. 2011. Wikipedia.  
<[http://en.wikipedia.org/wiki/Reverse\\_engineering](http://en.wikipedia.org/wiki/Reverse_engineering)> Luettu 3.11.2011.



## LSB:n muutoshistoriaa aikajanan muodossa



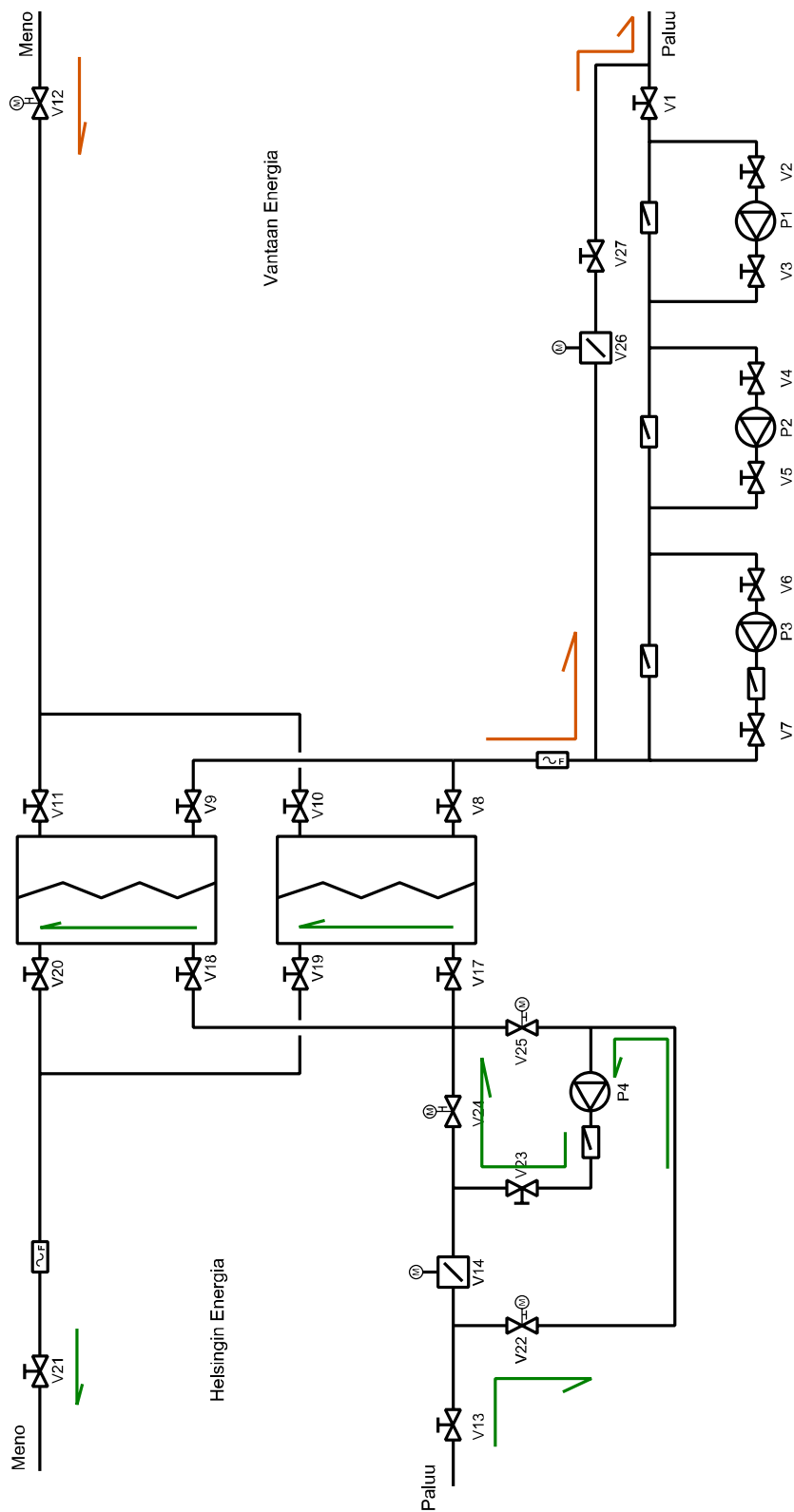
LSB:n kl-veden virtausreitit, kun kaukolämmön siirtosuuntana on HE:lta VE:lle



→ KL-veden virtaus normaalitilanteessa  
 → KL-veden virtaus, kun luonnollinen virtaama ei riitä vaaditulle siirtoteholle.  
 → Ostajan KL-veden virtaus

KL-energian siirto Helsingin Energialta Vantaan Energialle - LSB		SIVUT: 1 / 1
<b>VANTAAN ENERGIA</b>		
PIIRTÄNYT: L.V.		TILA: Julkaistu
PIIRRETTY: 2011		MUOKATTU:-

LSB:n kl-veden virtausreitti, kun kaukolämmön siirtosuuntana on VE:lta HE:lle



→ Ostajan tehostettu KL-veden virtaus  
→ Myyjän KL-veden virtaus

KL-energian siirto Vantaan Energialta Helsingin Energialle - LSB		SIVUT: 1 / 1
<b>VANTAAN ENERGIA</b>		PIIRRETTY: 2011
PIIRTÄNYT: L.V.	TILA: Julkaistu	MUOKATTU:-

Esimerkki LSB:n Simatic S5 -ohjelman tulkinnasta.

Tehonrajoituksen asettavan ohjelmakoodin lähtötilanne AWL-esitystavassa.

SEGMENT 5

```
-----  
      U      A 8. 7  
      SPB    =M017  
      A      DB 10  
      L      DD 42  
      L      DD 20  
      -G  
      T      DD 212  
      L      KG 15  
      XG  
      T      DD 214  
      L      DD 20  
      <G  
M017: =      M 56. 0  
      ***
```

SEGMENT 7

```
-----  
      U      A 8. 6  
      SPB    =M019  
      A      DB 10  
      L      DD 76  
      L      KG 94  
      >G  
      =      M 56. 1  
      U      M 56. 0  
      U      M 56. 1  
      =      M 56. 2  
      U      M 56. 2  
      =      A 8. 2  
M019: ***
```

Käsin kommentoituna liitteen 5 ensimmäisen sivun koodi.

FB/4

Segment 5:

**H-V ΔF \* 15 < FT-II**

label	Instruc	Operand	Comment
U	A 8.7	DS V → H AJOTAPA OHJAUS AKTIIVIEN	
SPB	=M017	MENE M017	
A	DB 10		
L	DD 42		(H:n virt.) - (V:n virt.)
L	DD 20		FT-41 - FT-11
-G			ERDUS DD12
T	DD 212		
L	KG 15		df = 15
XG			TULOS DD214
T	DD 214		
L	DD 20		JOS (df * 15) < V:IN VIRTAVS FT-II
<G			
M017: =	M 56.0		ASETA MERKKI M56.0
R ***			HUOM! ASETA M56.0 1, Koska kyse oli Tosi

Segment 6:

Segment 7:

**TEHON RAJOITUS VIRTAUKSESTA, KUN V → H**

label	Instruc	Operand	Comment
U	A 8.6	DS AJOTAPA H → V ON VALITTU	
SPB	=M019	MENE OHI	
A	DB 10		
L	DD 76		JOS DP76 > 94 (%)
L	KG 94		
>G			AKTIVOI M56.1 MUUS. TAP. FALSE
=	M 56.1		
U	M 56.0		
U	M 56.1		
=	M 56.2		
U	M 56.2		
=	A 8.2		

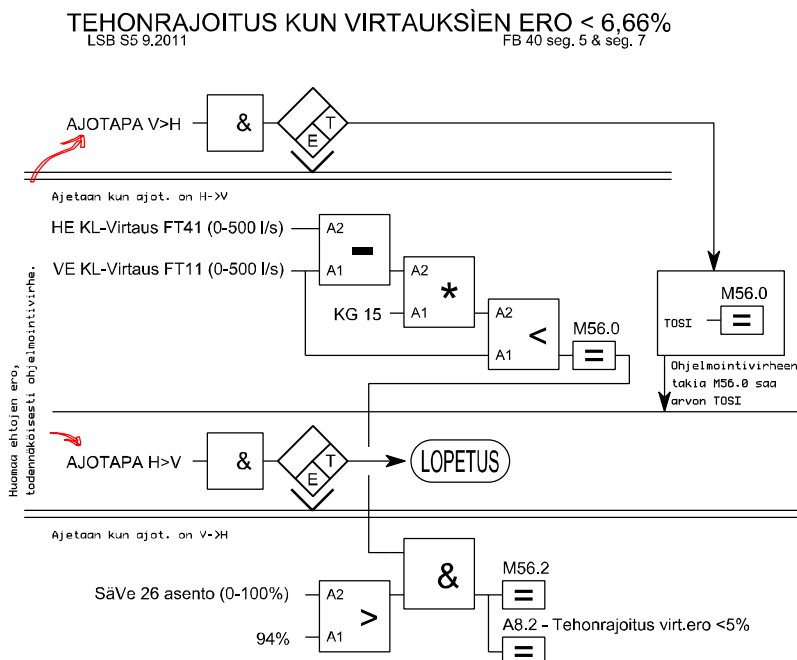
M019: \*\*\*

Segment 8:

SAATAJAN OHJEARVO  
CMP >G

TEHON RAJ. ΔF

Sovelletun FBD-esitystavan mukainen esitys liitteen 5 sivun 1 koodista selityksineen.  
(LSB S5 -Logiikkaohjelman toiminta 2011 – selvitys: s.42)



Ohjelman tarkoituksena on ollut mitä ilmeisimmin tarkistaa, onko KL-virtauksien ero alle 6,66 % verrattuna VE:n kl-virtaukseen (Myyjän) eli siis onko HE:n kl-virtaus (Ostajan) vähintään 6,66 % suurempi kuin vastaavasti VE:n kl-virtaus. Tämä havainto johtaa-kin olettamukseen, että molempien ajosuuntien virtauserojen tehonrajoituksesta hulehtivien ohjelmien alkuosat ovat menneet ristiin keskenään, koska säätöventtiileiden ohjauksesta vastaava ohjelma pyrkii pitämään myyjän virtauksen 7½ % ostajan kl-virtausta suurempana.

Tämä kyseisenkaltainen ohjelmointivirhe olisi mahdollista sotkemaalla M56.0 ja M55.0. Tällöin myös ajotavan tarkistukset menisivät kohdalleen.

Ohjelma siis tarkistaa onko HE:n virtaus vähemmän kuin 6,66 % VE:n virtausta suurempi, ja mikäli näin on, asettaa se merkkerin M56.0 todeksi. Tämän jälkeen on tarkistettu vielä, onko säätöventtiiliin V26 (VE) asento yli 94 %.

Lisäksi on ensimmäisen osan tarkistuksen lopetuksessakin tapahtunut toinen ohjelmointivirhe, sillä M56.0 saa arvon ajotavan kyselyn tuloksesta, joka siis on tosi, koska ohjelman ensimmäinen osa lopetetaan. Tämä virhe johtaa ajotavalla V->H tehonrajoitukseen aina, kun säätöventtiili V26 on yli 94 % auki.

LSB:n autostop-toiminto vuokaavioesityksenä

