Eerik Kelhä

VERSIONHALLINTAJÄRJESTELMÄ PLC- LÄHDEKOODEILLE

Sähkö- ja automaatiotekniikan koulutusohjelma
2019

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

VERSIONHALLINTA JÄRJESTELMÄ PLC- LÄHDEKOODEILLE

Kelhä, Eerik
Satakunnan ammattikorkeakoulu
Sähkö- ja automaatiotekniikan koulutusohjelma
Helmikuu 2019
Ohjaaja: Suvela, Timo
Sivumäärä: 43
Liitteitä: 1

Asiasanat: Versionhallinta, Versionhallintaohjelmistot, Ohjelmoitava logiikka, PLC, Lähdekoodi

---

Opinnäytetyön tarkoituksena oli selvittää, mitä olemassa olevista versionhallinta järjestelmistä soveltuu parhaiten PLC- lähdekoodeille. Työ tehtiin Cimcorp Oy:lle, koska Cimcorpilla ei ollut PLC- lähde-koodeille versionhallintajärjestelmää. PLC- lähdekoodit tallennettiin verkkolevyille tai muuhun dokumenttien hallintajärjestelmään.

Tavoitteenani oli tutkia ja vertailla olemassa olevia PLC- lähdekoodien versionhallintajärjestelmiä (mm. Rockwell, Siemens, Novotek, Bosch Rexroth) ja ottaa valittu järjestelmä käyttöön Cimcorpilla. Versionhallintajärjestelmän ominaisuudet ja niiden toiminnallisuus muiden ohjelmistojen kanssa oli tarkkailun ja testauksen kohteena, kuin myös versioinnin ja samassa projektissa työskentely usean henkilön kanssa samanaikaisesti.

Opinnäytetyön tiedot on kerätty yrityksen sekä valmistajien verkkosivuilta ja materiaaleista. Tehtävänäni oli myös laatia selkeä ohjeistus versionhallintajärjestelmälle, joka kattaa yksinkertaisen asennus- ja käyttöohjeen peruskäyttöä varten.

Tuloksena valittiin Cimcorp Oy:lle versionhallintajärjestelmä ja selkeät ohjeet sen asennukseen ja peruskäyttöön. Tutkituista versionhallintajärjestelmissä ei ole mahdollisuutta versioida samanaikaisesti projekteja osissa. Versiondog valittiin testaukseen siksi, koska se toimii monen ohjelmiston ja valmistajan kanssa yhteen, mitä Cimcorpillakin käytetään.

VERSION CONTROL SYSTEM FOR PLCS SOURCE CODES

Kelhä, Eerik
Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences
Degree Programme in electrical and automation engineering
February 2019
Supervisor: Suvela, Timo
Number of pages: 43
Appendices: 1

Keywords: Version control, Version Control Software, Programmable Logic Controller, PLC, Source code

---

The purpose of the thesis was to find out what existing version control system could be used for PLC source codes. The work was done for Cimcorp Oy because Cimcorp did not have a version control system for the PLC source codes. PLC source codes were stored on network disks or other document management systems.

My goal was to study and compare existing PLC source code management systems (including Rockwell, Siemens, Novotek, Bosch Rexroth) and to implement the selected system with Cimcorp. The features of the version management system and their functionality with other software were subject to monitoring and testing, as well as working with multiple programmers at the same time with the same project.

The thesis information has been gathered from the company's and manufacturers websites and from the material waves. My task was also to develop a clear guide to the version control system, which includes a simple installation manual.

As a result, Cimcorp Oy received a version management system and clear instructions for its installation and basic use. Not even one of the version control systems have the ability to run projects in parts at the same time, But Versiondog works with many other software and vendors, which led to Versiondog being tested and chosen.

SISÄLLYS

1 JOHDANTO

Opinnäytetyössä selvitetään, mitä olemassa olevaa versionhallintajärjestelmää voitaisiin käyttää PLC- lähdekoodeille. Tämä opinnäytetyö on tehty Cimcorop Oy:lle. Se on suuri järjestelmätoimittaja, jolla on toteutuneita projekteja 30 eri maassa. Cimcorp Oy on optimoinut lähettämöjä, jakelukeskuksia ja rengastehtaiden materiaalivirtoja yli 30-vuoden ajan, ja on yksi maailman rengasteollisuuden automaatiojärjestelmien toimittajista maailmassa. (Cimcorp www-sivut 2018.)

Cimcorpilla ei ollut PLC- lähdekoodeille versionhallintajärjestelmää, vaan lähdekoodit tallennetaan verkkolevyille tai muuhun dokumenttien hallintajärjestelmään. Koodia on tilanteesta riippuen monessa eri paikassa Cimcorpin järjestelmissä. Cimcorp haluaisi versionhallintajärjestelmän, jolla voisi tehdä yhtä projektia useampi ohjelmoija.

Tavoitteenani oli tutkia ja vertailla olemassa olevia PLC- lähdekoodien versionhallintajärjestelmiä, kuten esimerkiksi Rockwell, Siemens, Novotek, Bosch Rexroth ja ottaa valittu järjestelmä käyttöön Cimcorpissa. Tehtävänäni oli myös laatia versionhallintajärjestelmälle selkeä ohjeistus, joka kattaa yksinkertaisen asennus- ja käyttöohjeen peruskäyttöä varten. (Cimcorp Oy 2018.)

Työssä tutkitaan ja vertaillaan versionhallintajärjestelmiä ja niiden ominaisuuksia. Lisäksi työssä tutustutaan PLC-ohjelmointiin, versionhallintaan ja lähdekoodin perusteisiin.

2 YRITYS

2.1 Cimcorp Oy

Cimcorp on yritys, joka on sisälogistiikan uranuurtaja ja on robottipohjaisen logistiikan automaatioratkaisujen edelläkävijä. Cimcorp pyrkii yksinkertaistamaan tehdas- ja lähettämönmateriaalin liikkuvuutta, hyödyntäen ja suunnitellen älykkäitä ohjelmistoja ja robottiratkaisuja. Cimcorp Oy on tunnettu suurien portaalirobottien valmistaja rengas- ja elintarviketeollisuudelle. (Cimcorp www-sivut 2018.)

Kuva 2.1 (Cimcorp Oy, Ulvilassa sijaitseva pääkonttori.)

Cimcorp on optimoinut lähettämöjä, jakelukeskuksia ja rengastehtaiden materiaalivirtoja yli 30-vuoden ajan, ja on yksi maailman johtavista konserneista, jotka toimittavat rengasteollisuuden automaatiojärjestelmiä ympäri maailman. Toteutuneita projekteja on 30 eri maassa. Suomessa Cimcorp on vahvoilla myös elintarviketeollisuuden ja vähittäiskaupan jakelun keräilyjärjestelmien sekä postinjakelun automaation toimittajana. (Cimcorp www-sivut 2018.)

Yrityksellä on pääkonttori Ulvilassa sekä tytäryhtiöt Kanadassa ja Yhdysvalloissa. Suomessa olevat toimipisteet tarjoavat huoltoa Helsingissä, Lahdessa ja Jyväskylässä. Cimcorp-konserni on japanilaisen Murata Machinery, Ltd:n omistuksessa. Tähän konserniin kuuluvat Cimcorp Oy Ulvilassa, Cimcorp Automation Ltd. Kanadassa ja Cimcorp USA, Inc. Yhdysvalloissa. (Cimcorp www-sivut 2018.)

Murata Machinery, Ltd. on viidenneksi suurin logistiikka-automaation toimittaja maailmassa, ja sen liikevaihto oli yli 1,6 miljardia vuonna 2017. Cimcorp Oy:n liikevaihto oli vuonna 2017 87,1 miljoonaa ja on noussut reilusti vuodesta 2015 lähtien, jolloin liikevaihto oli 55,1 miljoonaa. (Cimcorp www-sivut 2018.)

2.2 Historia

Vuonna 1986 yrityskaupan myötä Rosenlewin työkalutehtaan automaatio-osastosta tuli Wärtsilän tytäryhtiö Cimcorp Oy, kun vielä vuonna 1975 Cimcorp Oy oli aluksi Porilaisen Oy W.Rosenlew Ab:n työkalutehtaan automaatio-osasto, jonka pääpaino oli robotiikassa. Kehitys yrityksen sisällä lähti nousuun Valcon kuvaputkitehtaan suurtilauksesta, jolloin Rosenlew toimitti kaikki mahdolliset koneet ja robotit Valconille. Tilauksen myötä yritys eteni monille muille teollisuuden aloille. (Cimcorp www-sivut 2018.)

Yrityskauppoja tehtiin jälleen vuonna 1996, jolloin Cimcorp Oy siirtyi Swisslogin omistukseen ja jonka myötä kesällä 2002 Cimcorp Oy vaihtoi nimeään Swisslog Oy:ksi.

Loppuvuodesta 2003 lähtien Swisslog Oy oli siirtynyt toimivan johdon omistukseen, jonka jälkeen 1.1.2004 alkaen robotiikan yritys oli jälleen muuttanut nimensä Cimcorp Oy:ksi. (Cimcorp www-sivut 2018.)

2000-luvun alussa elintarviketeollisuuden ja autonrenkaiden käsittelyn automatisoinnin myötä liiketoiminta kasvoi entisestään. Cimcorp alkoi tuolloin olla myös tunnettu suurikokoisista portaalirobotteistaan. Järjestelmät kasvoivat suuriksi, ja Cimcorp oli valmis omilla varastonhallinta- ja valmistuksenohjausohjelmistoillaan antamaan oman vastauksen kehitykseen ja ratkaisuihin. (Cimcorp www-sivut 2018.)

Vuonna 2010 Cimcorp Oy osti itselleen kanadalaisen robottivalmistajan RMT Roboticsin, kaupan jälkeen 1.1.2015 tytäryhtiö muutti nimensä Cimcorp Automation Ltd:ksi. (Cimcorp www-sivut 2018.)

Vuonna 2014 tapahtui jälleen omistajan vaihdos Cimcorp Oy:ssä, kun japanilainen Murata Machinery Ltd osti koko osakekannan 30.10.2014. Omistuksen vaihdos ei vaikuttanut Cimcorpin nimeen, brändiin tai muuhun yrityksen osa-alueeseen, vaan jatkui liiketoiminallisesti ennallaan. (Cimcorp www-sivut 2018.)

2.3 Ulvilan pääkonttori

Urakkasopimus nykyisestä Ulvilan pääkonttorista solmittiin Rakennus Vuorenpää Oy:n kanssa 15.8.2001. Rakentamiskustannukset olivat kaiken kaikkiaan 43 miljoonaa markkaa. Aikataulun mukaan Logistiikkatalo olisi muuttovalmis 31.7.2002. (Cimcorp intranet 2018.)

Swisslog Oy ja Ulvilan kaupunki käynnistivät yhteisen teollisuustilahankkeen vuonna 2000 perustamalla Kiinteistö Oy Ulvilan Logistiikkatalon. Swisslog Oy:lle rakennettavan Logistiikkatalon peruskiven muuraustilaisuus järjestettiin 8. lokakuuta 2001 Swisslogin uuden toimitilan tontilla, Satakunnantie 5, Ulvila. (Cimcorp intranet 2018.)

29.11.2000 aloitettiin hankkeen toteutussuunnittelu ja tilojen arkkitehtinen suunnittelu, jonka pääsuunnittelusta vastasi Arkkitehtitoimisto Küttner Ky. Noin 300 hengelle mitoitettujen toimitilojen kokonaispinta-alaksi suunniteltiin 9800 neliömetriä. (Cimcorp intranet 2018.)

Kiinteistö Oy Ulvilan Logistiikkatalon harjannostajaiset pidettiin perjantaina 01.03.2002 Logistiikkatalon työmaalla, joka on nykyinen pääkonttori. Elokuussa 2002 oli Ulvilaan muutto ja käyttöönotto (Cimcorp intranet 2018.)

Vuonna 2018 Ulvilan pääkonttorilla aloitettiin laajennustyöt. Nykyisissä tiloissa (2018) on 5800 neliömetriä tuotantotiloja ja 3700 neliömetriä toimistotilaa. Laajennus laajentaa lattiapinta-alaa 7 900 neliömetriä, mikä lisää tuotannon tiloja 5 700 neliömetriä ja toimistojen tiloja 2200 neliömetriä. (Cimcorp intranet 2018.)

Tuotantotilaa kasvattamalla Cimcorp pystyy vastaamaan kasvaviin asiakastarpeisiin, koska tuotantokapasiteetti kasvaa, joten kyky hallita suurempia projekteja paranee. Lisäksi uusi kolmikerroksinen toimistorakennus on apu tulevaisuuden henkilöstökasvulle. Erityisesti laajennettu tuotanto mahdollistaa Cimcorpin vahvuuksien hyödyntämisen ainutlaatuisten materiaalinkäsittelyjärjestelmien kehittämisessä. Tällä laajennuksella Cimcorpin kokonaispinta-ala on 17 400 neliömetriä. (Cimcorp intranet 2018.)

3 PLC

3.1 Programmable Logic Controller

PLC on lyhenne englanninkielisestä termistä Programmable Logic Controller ja tarkoittaa ohjelmoitavaa logiikkaa. Logiikka on kuin pieni tietokone, joka pitää sisällään mikroprosessoreita. PLC on ohjelmoitava elektroninen ohjauslaite, joka digitaalisesti käyttää muistiaan. Siihen on tallennettu sekvenssit, ajoitukset ja laskennat, jotka suoritetaan suoraan muistiin kirjoitetusta ohjelmakoodista syklisesti, erittäin nopeasti jokainen käsky yksi kerrallaan (PLC-tutor ja PLCOpen www-sivut 2018.)

Kuva 3.1 (Siemens S7-1200, Logiikka/CPU, Siemens www- sivut 2018.)

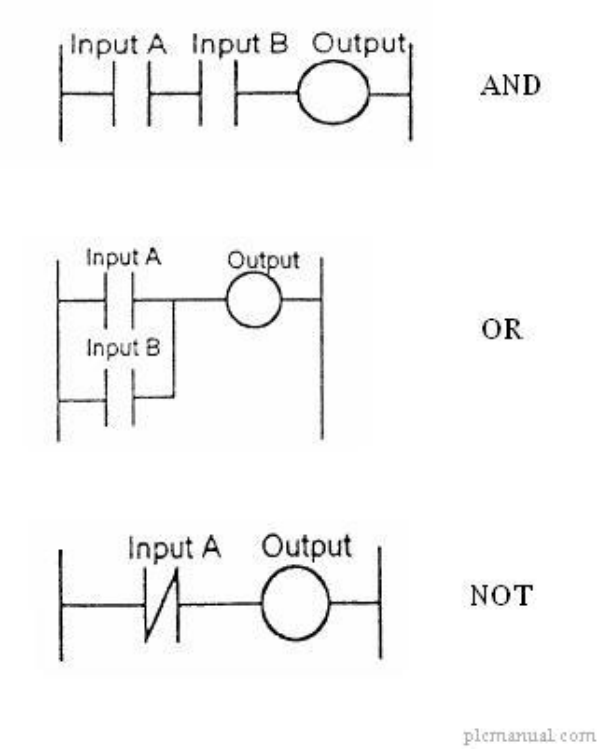
Logiikalla voidaan ohjata massiivisia ja monia eri automaation kokonaisuuksia samanaikaisesti, koska logiikka lukee kirjoitetun loogisen ohjelman useasti sekunnin aikana,

mutta käsky kerrallaan. Logiikka lukee myös jatkuvasti siihen kytkettyjen I/O tietojen tiloja eri laitteita apuna käyttäen. Logiikkaan voidaan yhdistää todella paljon eri komponentteja, kuten monia antureita. (PLC-tutor ja PLCOpen www-sivut 2018.)

PLC-järjestelmille on oma standardinsa (IEC 61131), joka tarkoituksena on yhdenmukaistaa kirjoitetun ohjelmien ja käskyjen samankaltaisuutta. Standardi määrittelee viisi ohjelmointikieltä, joita yleisesti käytetäänkin ohjelmoitavien logiikoiden ohjelmoimiseen. (Sesko www-sivut 2018.)

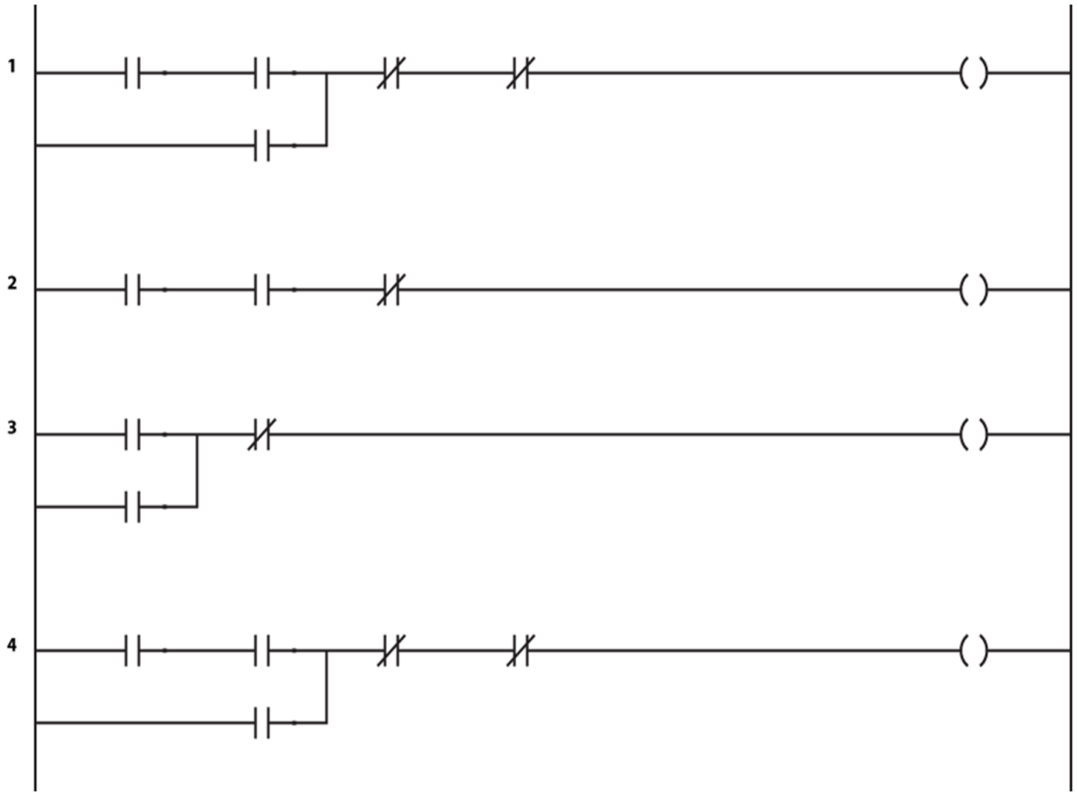
3.2 Ohjelmointikielet

Ohjelmoinnissa käytetään seuraavia ohjelmointiin tarkoitettuja kieliä: Ladder Diagram (LD), joka on tuttu vuosien taka, Function Block Diagram (FBD), Structure Text (ST), Instruction List (IL) sekä Sequential Function Chart (SFC). (PLC-Academy www-sivut 2018.)



Kuva 3.2 (LD, Ohjelmointikieli.)

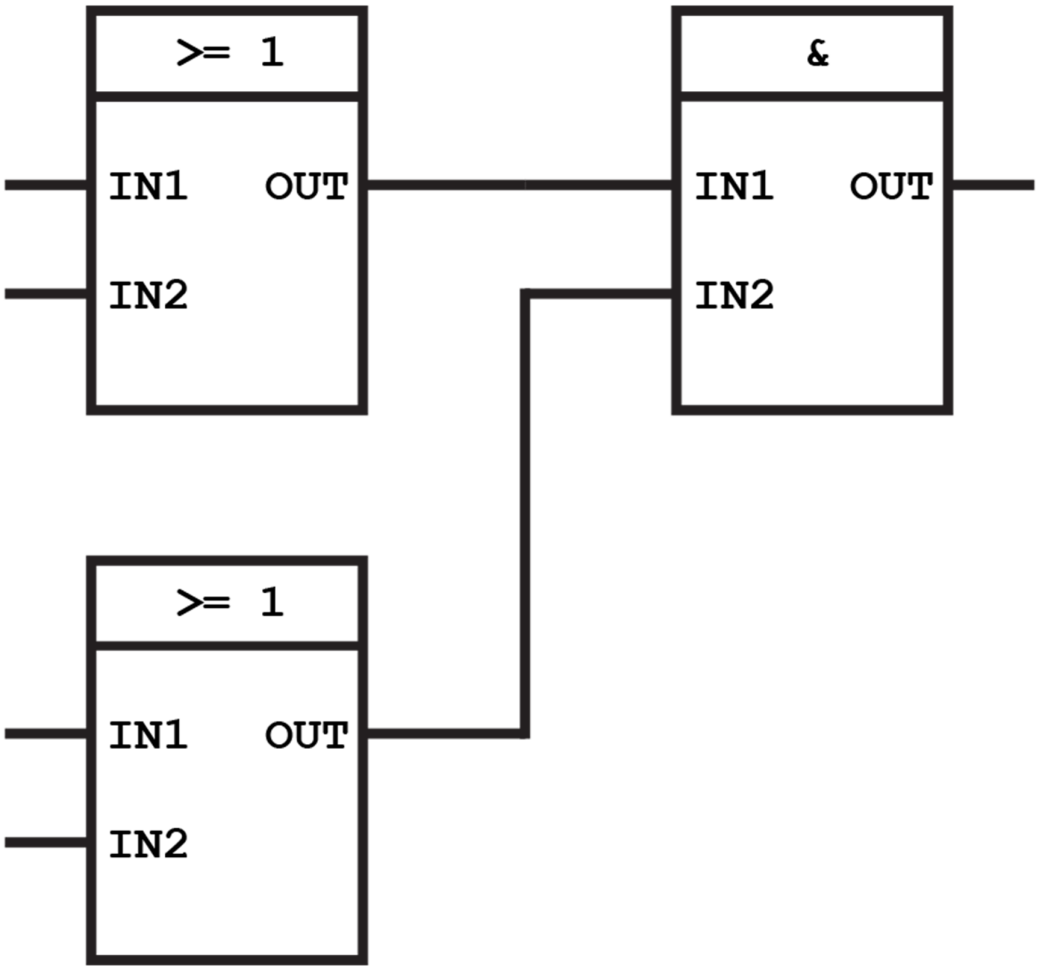
Ladder Diagram tunnetaan myös nimellä Ladder logic. Ohjelmointikielen nimi tulee suoraan siitä, että se muodostuu pylväistä sekä vaakaviivoista, jotka näyttävät vähän tikapuilta. Tämä ohjelmointikieli on pääasiassa tarkoitettu bittilogiikkatehtäviin, kuten yksinkertaisimpiin käskyihin. Esimerkiksi ”On” ja ”Off” käskyt. Ladder Diagram on standardisoitu IEC 61131-3. (PLC-Academy www-sivut 2018.)



Kuva 3.3 (LD, Ohjelmointikieli.)

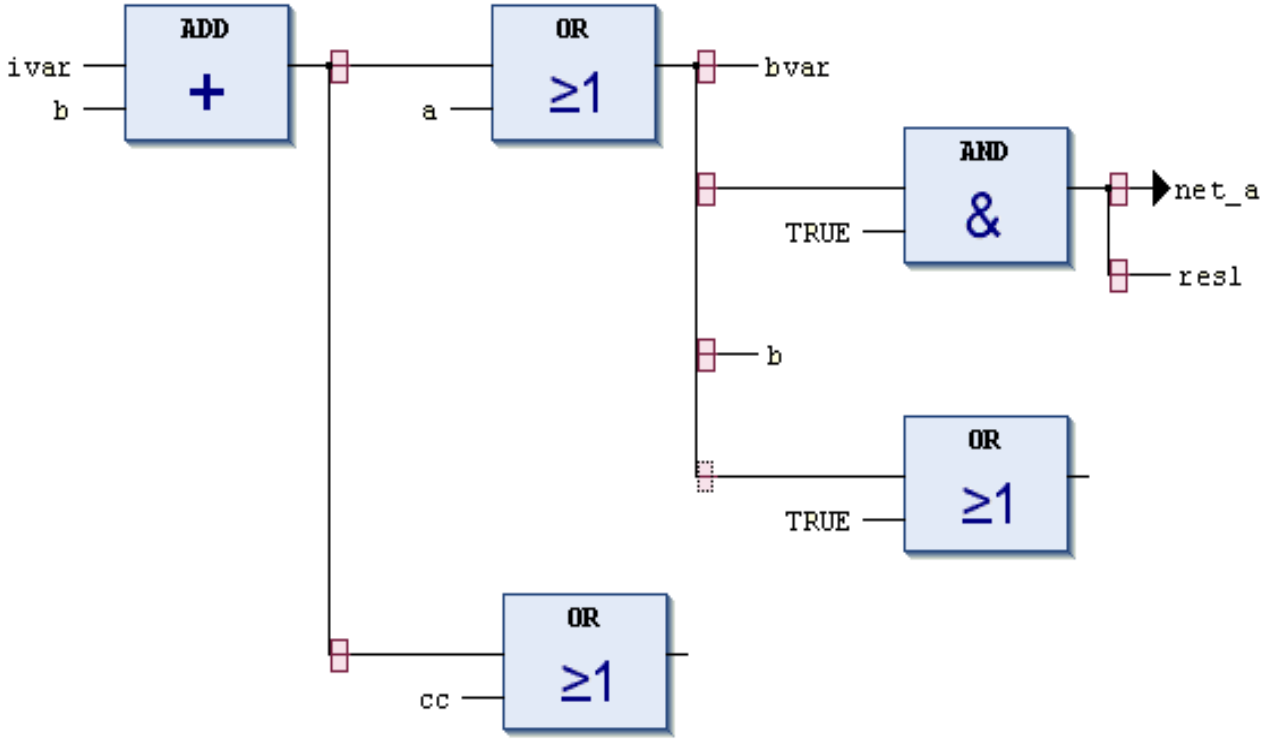
Function Block Diagram on yksinkertaisin ja graafisin tapa ohjelmoida. Function Block Diagram on helppo oppia, ja sen graafisen ohjelmoinnin avulla on mahdollisuus tehdä monimutkaisiakin ohjelmia helposti. Function Block Diagram löytyy myös standardista IEC 61131-3. (PLC-Academy www-sivut 2018.)

Function Block Diagram pitää sisällään erilaisia laatikoita, joita yhdistetään toisiinsa. Laatikot pitävät sisällään eri toimintoja, ikään kuin valmiiksi kirjotettu koodinpätkä, kuten erilaisia ajastimia, laskureita, vertailuja ja muuttujia. (PLC-Academy www-sivut 2018.)



Kuva 3.4 (FBD, Ohjelmointikeli. Vasemmalla olevat laatikot ovat OR ja oikealla puolella on AND funktio.)

Laatikoita kutsutaan siis Function block -nimellä, ja jokaisella Function block:illa on oma symboli, joka kertoo mitä kyseinen Function block tekee. Kaikki käskyt toteutuvat näiden laatikoiden avulla ja ne sisältävät aina vähintään yhden sisääntulon (Input) ja ulostulon (Output). (PLC-Academy www-sivut 2018.)



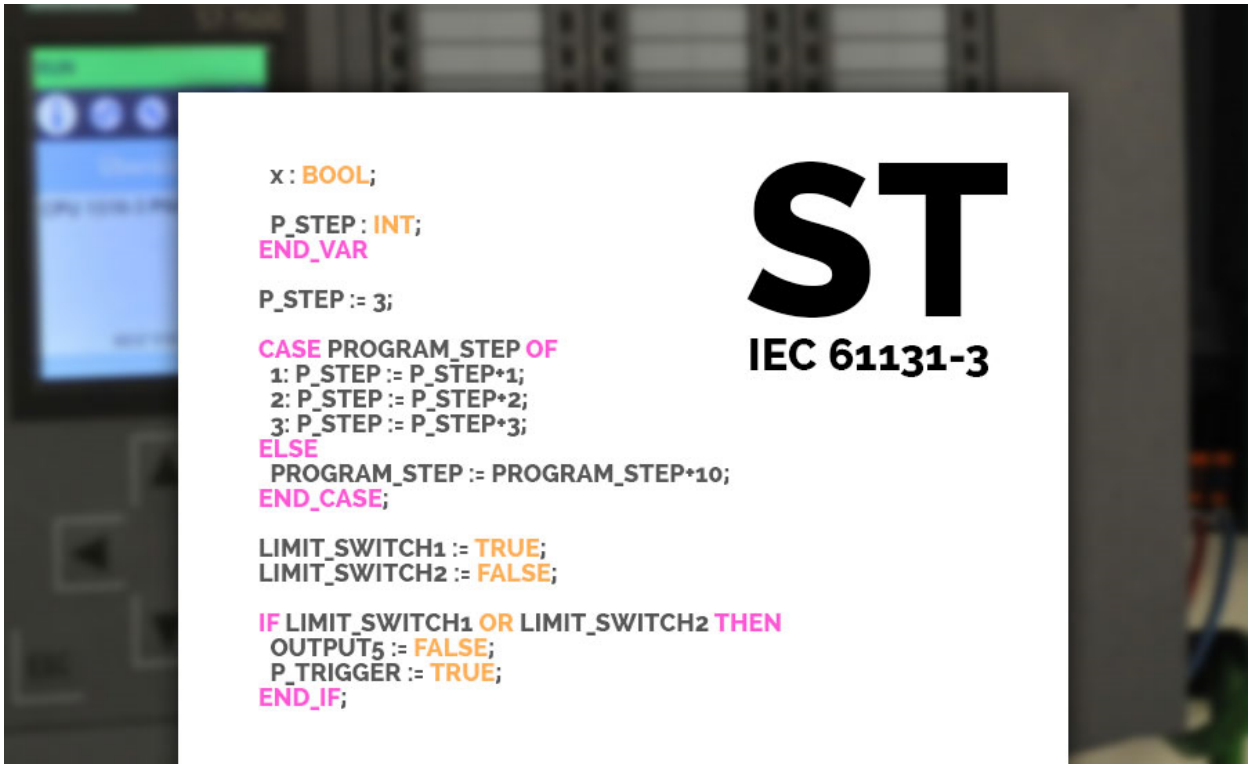
Kuva 3.5 (FBD, Ohjelmointikeli.)

Structure Text on toisten ohjelmoitsijoiden kannalta hyvä ohjelmointikieli, koska se on luettavaa tekstiä ja helpompi ymmärtää kuin FBD-ohjelmointikieli. Structure Text ohjelmointikielenä on tekstipohjaista, mikä kuuluu standardiin IEC 61131-3. (PLC-Academy www-sivut 2018.)

Kuva 3.6 (Structure Text toimii sykleissä, eli lukee ja suorittaa käskyjä uudelleen ja uudelleen kerta toisensa jälkeen.)

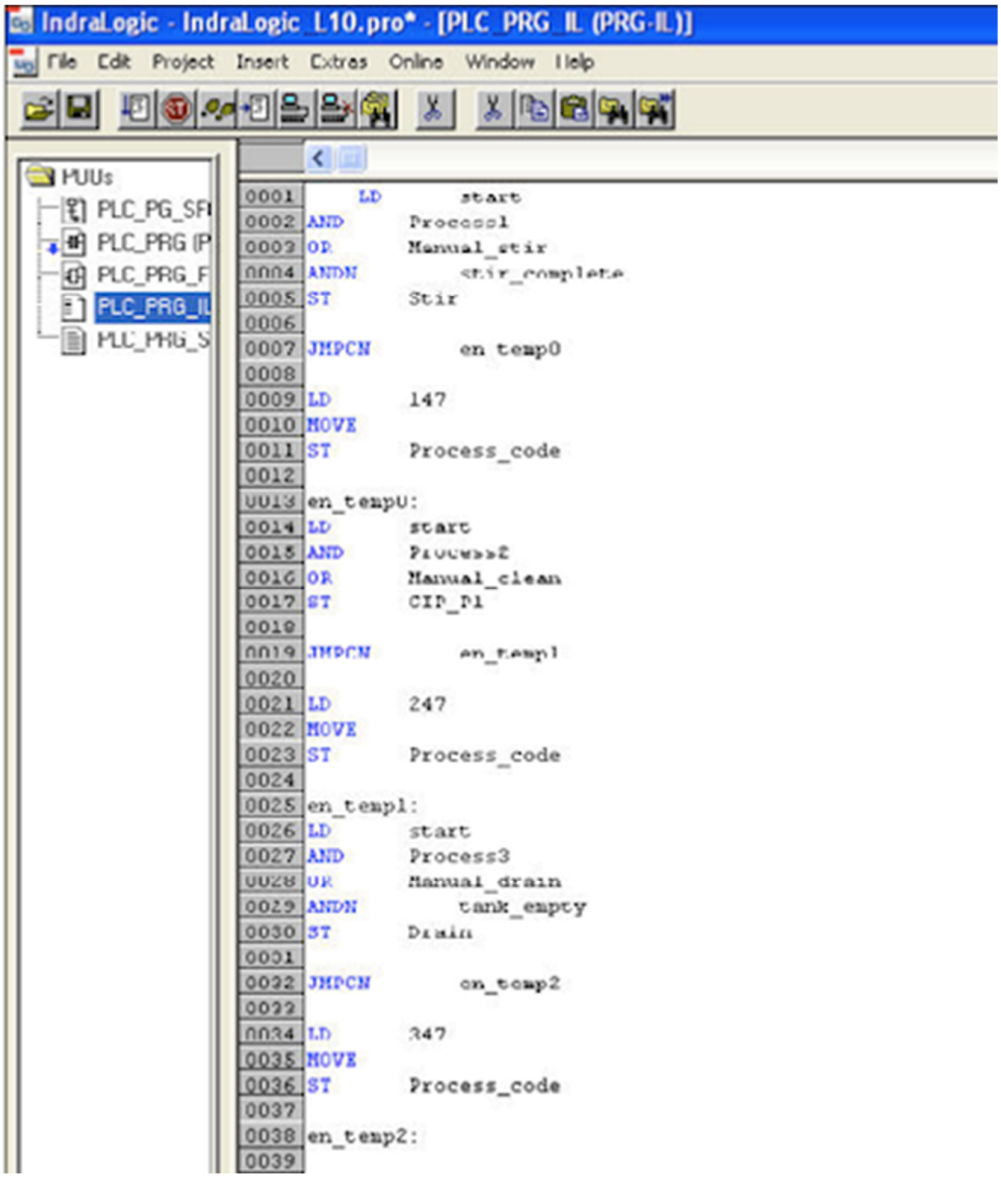
Esimerkiksi ohjelma alkaa termillä “PROGRAM” ja loppuu “END_PROGRAM”, kaikki edellä mainittujen sanojen välissä oleva teksti on PLC- ohjelmaa. Structure Text toimii kuten muutkin tietokoneen ohjelmointikielet. Ohjelmointi kielet sisältävät eri käskyjä, joiden kanssa ohjelma rakentuu. (PLC-Academy www-sivut 2018.)

Kuten esimerkkikuvassa (kuva 3.7) käskyt omaavat tietyn tarkoituksen ja muuttujat lauseiden sisällä muuttuvat ohjelman mentäessä eteenpäin. Tarkoituksena on saada aktivoitua haluamaa asiaa. Esimerkiksi tässä lasketaan syklien määrää. (PLC-Academy www-sivut 2018.)



Kuva 3.7 (Structure Text, Esimerkkikuva)

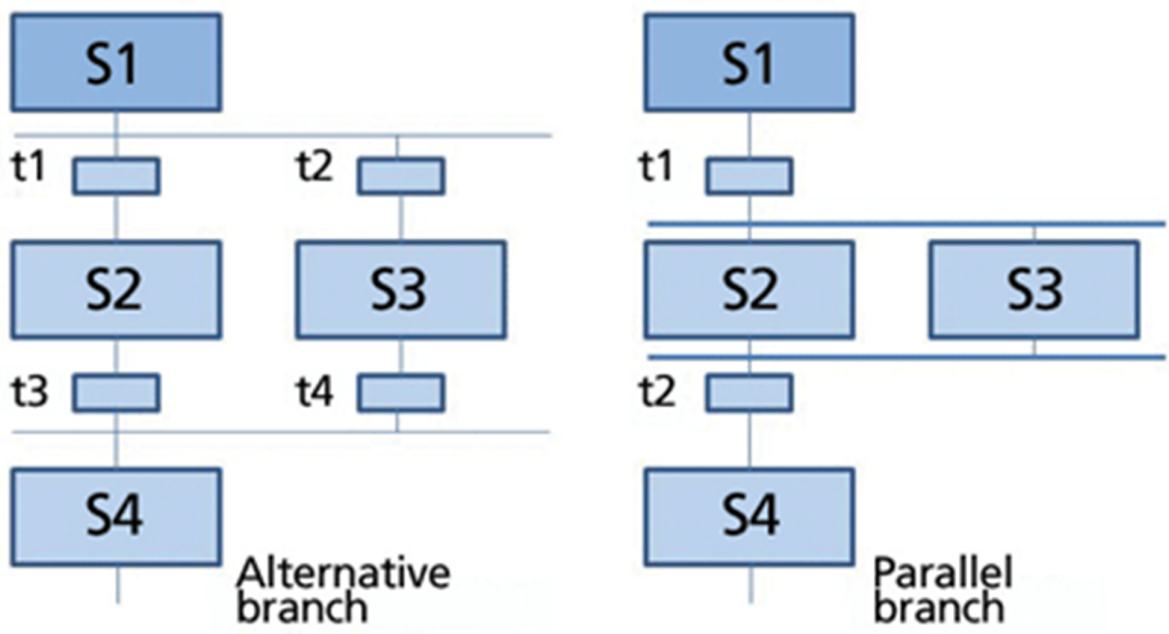
Instruction lists (tai IL) on yksi ohjelmointikielistä, joka kuuluu myös IEC 61131-3 standardiin. Esimerkiksi jotkin yhteiset toiminnot ovat matemaattisia, kuten arvojen lisääminen, vähentäminen, kertominen ja jakaminen. (PLC-Academy www-sivut 2018.)



Kuva 3.8 (Instruction list, Esimerkki)

Kuten kaikilla PLC-ohjelmointikielillä, IL:llä on etuja ja haittoja. Selkeimpiä etuja on ohjelman toteutusnopeus ja se vie vähän muistitilaa. Tämä on selvä etu erityisesti PLC:ssä, jossa on tiukasti muistitilaa. Haittapuolena on, että kieli ei ole yleinen, koska monet ihmiset suosivat graafisia ohjelmointikieliä ja -ympäristöjä. (PLC-Academy www-sivut 2018.)

Sequential Function Chart on kuin peräkkäiset funktiokaaviot (SFC) ja on yksi viidestä IEC 61131-3 -standardin määrittelemästä PLC-ohjelmointikielestä. SFC on graafinen ohjelmointikieli, ei tekstipohjainen. Tässä tapauksessa (Kuva 3.9) graafinen ohjelmointikieli tarkoittaa sitä, että se soveltuu hyvin suurten ja monimutkaisten prosessien hajottamiseen pienemmiksi paloiksi, joita on helpompi nähdä ja ymmärtää. (PLC-Academy www-sivut 2018.)



Kuva 3.9 (Sequential Function, esimerkkikaaviokuva)

Sequential Function:n taustalla olevat keskeiset käsitteet ovat askeleet ja siirtymät (”Steps” ja ”Transitions”). Vaihe (”Step”) on pohjimmiltaan jonkin verran funktio koko järjestelmässä, kuten yksittäisessä koneprosessissa. Siirtyminen (”Transitions”) on juuri sitä, siirtymistä vaiheesta toiseen. (PLC-Academy www-sivut 2018.)

3.3 Historia

Vuonna 1968 vahvistettiin ohjelmoitavan logiikan kriteerit kuten se, että laitteen pitää selviytyä teollisessa ympäristössä ja teollisuuden vaikutuksen alaisena. PLC pitää olla helposti ohjelmoitavissa insinööreille ja asentajille. Laitteen pitää olla myös uudelleen ohjelmoitavissa ja kokonaan uudelleen asennettavissa uudelle prosessin vaiheelle. Vuonna 2018 ohjelmoitava logiikka omaa yhä nämä kriteerit. (PLC-Tutor www-sivut 2018.)

Ensimmäisiä ohjelmoitavia logiikoita oli Modular Digital Controller, MODICON. Tämä on edelleen suosiossa oleva logiikka muiden joukossa, kuten Siemens ja Omron. PLC oli ensimmäisenä autoteollisuudessa käytössä. Autoteollisuus halusi helpon ohjelmointitavan heidän insinööreille ja asentajilleen. Tuloksena syntyi Ladder Logic eli niin sanottu tikapuulogiikka. Ladder Logic on hyvin yksinkertainen ohjelmointitapa ohjelmoitavalle logiikalle, joka on hyvin samanlainen lukea kuin relelogiikoiden kytkentäkaaviot. Tämä ”tikapuukieli” on edelleen suosittu ohjelmointikielimalli, vaikka sen rinnalle on kehitetty monia muita malleja ja varsinaisia ohjelmointikieliä. (PLC-Tutor www-sivut 2018.)

4 LÄHDEKOODI

4.1 Lähdekoodit

Lähdekoodilla tarkoitetaan ohjelman luettavaa rakennetta, jonka ohjelmoija on kirjoittanut jollakin ohjelmointikielellä. Tekstimuodossa olevan ohjelman ihminen pystyy lukemaan ja ymmärtämään. Tietokoneet tai logiikat eivät pysty suoraan toimimaan ohjelmoijan tekstitiedoston avulla, vaan tietokone ja logiikka tarvitsevat tekstitiedoston käännettynä binääriseen hyvin yksin kertaiseen muotoon. ”Käännettyä koodia kutsutaan objektikoodiksi”. Ihmiselle binäärikoodin lukeminen on mahdotonta, koska se koostuu ykkösistä ja nollista. (Techopedia www-sivut 2018.)

Lähdekoodin voi kuka tahansa kirjoittaa, jos tietää millä kielellä ohjelmaa alkaa ohjelmoida. ”Lähdekoodi on tietokoneohjelman lähde”. Valmis ohjelma, kuten Microsoft Office –ohjelmistot pitävät sisällään useampaa lähdekoodista käännettyjä tiedostoja, jotka muodostavat yhdessä toimivia ohjelmistoja. (Techopedia www-sivut 2018.)

Lähdekoodi voi pitää sisällään erilaisia käskyjä, toimintoja, ohjeita, ja silmukoita joiden avulla ohjelmoija on suunnitellut ohjelman toimivan. Ohjelma käännetään ja simulaation avulla voidaan heti todeta, toimiiko haluttu koodi, ja jos se toimii, niin voidaan ohjelma tallentaa koneen muistiin vielä muokattavaksi tai suoraan käyttöön. (Techopedia www-sivut 2018.)

4.2 PLC lähdekoodi

Ohjelmoitavissa logiikoissa lähdekoodi eli varsinainen ohjelmointi tapahtuu tietokoneen tai jonkin muun laitteen avulla. Logiikassa ei itsessään ole kirjoitusohjelmia tai muita kääntäjiä. Ohjelma kirjoitetaan PC:hen asennetulla ohjelmalla. (Techopedia www-sivut 2018.)

Ohjelmoitavissa logiikoissa on valmistajakohtaisesti tietyt ohjelmointiympäristöt, joilla koodi kirjoitetaan, mutta käytännössä ohjelmointikieli on sama jokaisella. Ohjelmointi tapahtuu standardi IEC 61311-1 ja IEC 61311-3 mukaisesti. (Techopedia www-sivut 2018.)

```
11  END_IF
12
13  IF (IN1=0 AND AUTO=0) OR (AUTO AND IN2=0) THEN
14      SET2:=1;
15  END_IF
16  IF EMGCY OR (OUT2 AND SET2 AND START) OR (SET2 AND OUT2=0) THEN
17      SET2:=0;
18  END_IF
19
20  Timer01(PT:=Value01*1000, IN:=SET1);
21  Timer02(PT:=Value02*1000, IN:=SET2);
22  Rise02(CLK:=Timer01.Q);
23  Rise03(CLK:=Timer02.Q);
24
25  IF Rise02.Q THEN
26      SET3:=1;
27  END_IF
28  IF Rise03.Q OR EMGCY THEN
29      SET3:=0;
30  END_IF
31
32  OUT1:=EMGCY=0 AND AUTO=0 AND OUT2=0 AND MANUEL;
33  OUT2:=EMGCY=0 AND OUT1=0 AND SET3;
```

Kuva 4.1 (ST, ohjelmointikieli, hyvin samanlainen kuin muut tekstimuodossa olevat ohjelmointikielet.)

5 PLC –OHJELMOINTI

5.1 Ohjelmakomponentit

Tässä viitataan Standardiin IEC61131-3, mikä pyrkii yhtenäistämään ohjelmoitavien logiikoiden ohjelmoinnissa suoritettavat toiminnot, kuten: muuttujien määrittely, ohjelman rakenne ja ohjelmointikielet. Standardin tavoitteena on, että ohjelmistoista tulisi laitevalmistaja riippumaton eli esimerkiksi Siemensin ohjelmoitavaan logiikkaan tehty ohjelma voitaisiin siirtää jonkun toisen valmistajan ohjelmoitavaan logiikkaan esimerkiksi Omronin PLC:hen. (Automaatiotekniikka/Timo Suvela 2011.)

Yleiset komponentit ohjelmoitavassa logiikassa, esimerkiksi Siemens, ovat Funktio (FC) ja Toimilohko (FB), joita käytetään ohjelmoidakseen tiettyä toiminnallista kokonaisuutta. Näitä voidaan kutsua rajattomasti ohjelmassa. Pääohjelmana Siemensillä toimii ohjelmalohko OB1. (Automaatiotekniikka/Timo Suvela 2011.)

Funktio (FC) ei sisällä sisäistä muistia, joten funktioon syötetyt tulot vaikuttavat suoraan funktiosta lähteviin arvoihin, ja tästä syystä se soveltuu huonosti monimutkaisten toimintojen toteuttamiseen. Funktio sisältää ohjelmakoodia ja tulo- ja lähtöparametrien määrittelyn. FC:tä kutsuvan sovelluksen ja funktion välinen tiedonsiirto suoritetaan tulo- ja lähtöparametrien kautta. Funkitota (FC) on mahdollista käyttää myös ohjelman jakamisessa pienempiin osakokonaisuuksiin, jolloin ohjelmien rakenne ja sitä kautta myös luettavuus paranevat. Samaa koodia ei tarvitse kirjoittaa joka paikkaan, vaan voi käyttää samaa funktiota useassa eri paikassa. (Automaatiotekniikka/Timo Suvela 2011.)

Funktion (FC) kaikki ohjelmassa käytettävät muuttujat määritellään lohkoissa seuraavasti: (IN) Tuloparametrit, (OUT) Lähtöparametrit, (IN_OUT) Tulo – lähtöparametrit ja (TEMP) Lohkon sisäiset, väliaikaiset muuttujat, jotka nollautuvat, kun funktion käsittely loppuu. Funktiossa käsiteltävät arvot siirretään kutsuvasta ohjelmasta funktioon IN- ja IN_OUT- parametrien kautta. Funktio palauttaa päivittämänsä arvot kutsuvaan

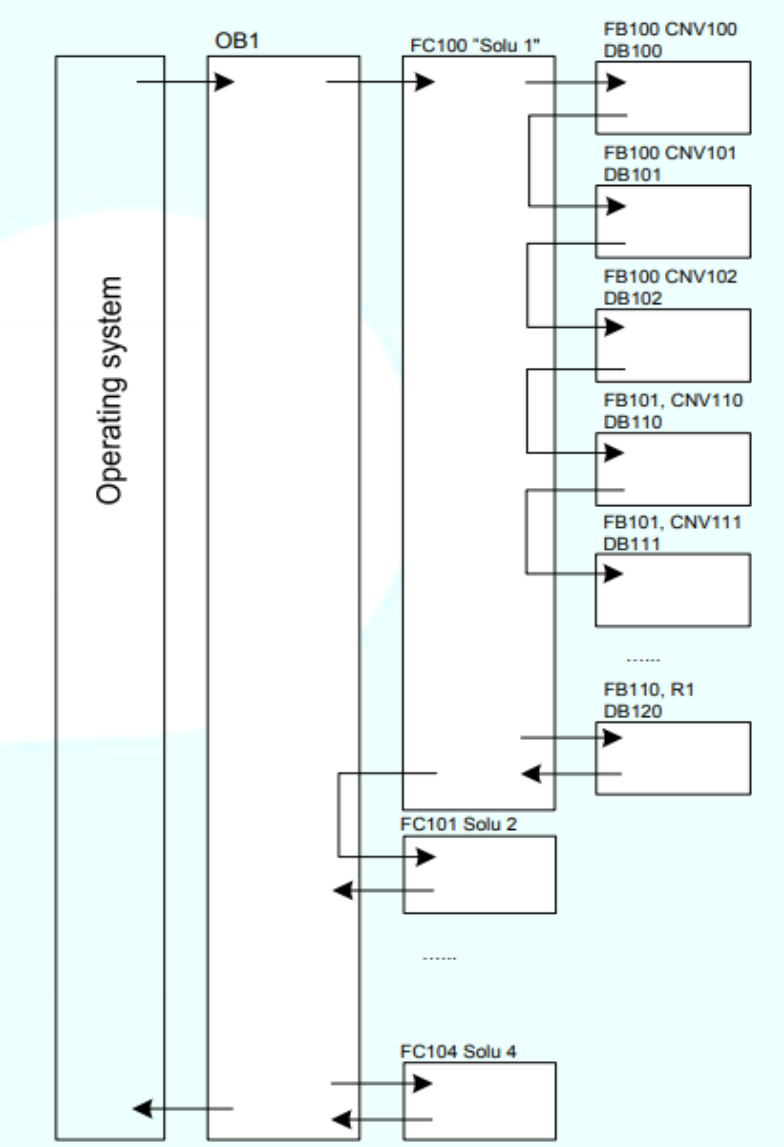
ohjelmaan OUT – ja IN_OUT parametrien kautta. (Automaatiotekniikka/Timo Suvela 2011.)

IN - kohtaan määritellään tuloparametrit, eli ne muuttujat, joiden alkuarvo halutaan määritellä kutsuvassa ohjelmassa. Funktion sisällä ei ole mahdollista muuttaa tuloparametreihin määriteltyjä arvoja. OUT - kohtaan määritellään lähtöparametrit, eli ne muuttujat, joiden arvo halutaan palauttaa kutsuvaan ohjelmaan. IN_OUT kohtaan määritellään parametrit, joissa tieto siirtyy molempiin suuntiin. Tällaisten muuttujien arvoja voidaan päivittää sekä funktiossa, että funktion ulkopuolella sovelluksessa. (Automaatiotekniikka/Timo Suvela 2011.)

Toimilohko (FB) on toiminnaltaan lähes samanlainen kuin funktio, paitsi, että toimilohko varaa itselleen muistia, jolloin sitä voidaan soveltaa ohjelmassa funktiota monipuolisemmin. Toimilohkoon määriteltyjen sisäisten muuttujien tilat
säilyvät lohkon muistissa lohkon suorituksen jälkeenkin, jolloin niitä voidaan hyödyntää, kun lohko seuraavan kerran suoritetaan. Esimerkiksi Siemens S7 ohjelmissa toimilohkon sisäiset muuttujat talletetaan erilliseen instanssitiedostoon. Ohjelmaeditori muodostaa instanssitiedoston automaattisesti, kun toimilohko liitetään ohjelmaan. Instanssitiedoston muuttujat, niiden nimet ja tyyppi määritellään toimilohkossa. Esimerkiksi toistuvista toiminnoista, kuten kuljettimien toiminnallisuudesta kannattaa tehdä toimilohkolla. (Automaatiotekniikka/Timo Suvela 2011.)

5.2 Ohjelma-arkkitehtuuri

Funktiota (FC) ja Toimilohkoja (FB) on mahdollista käyttää myös ohjelman jakamisessa pienempiin osakokonaisuuksiin, jolloin ohjelmien rakenne ja sitä kautta myös luettavuus paranevat. Laiteohjauksen arkkitehtuurista johtuen, tämä tapa on yleisesti käytetty. Jokaisen laitteen ohjaus sijoitetaan omiin ohjelmalohkoihinsa, jotka ryhmitellään fyysisen sijainnin perusteella omiksi kokonaisuuksikseen. (Automaatiotekniikka/Timo Suvela 2011.)

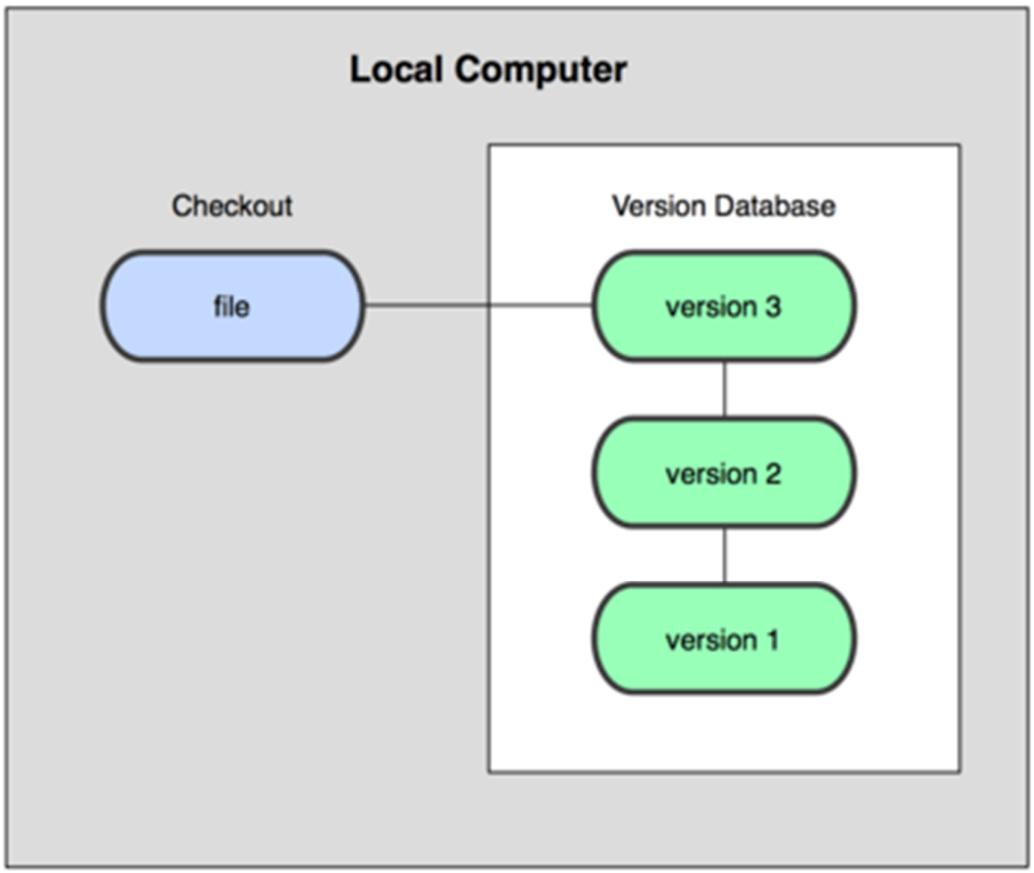


Kuva 5.1 Kuvassa on esitetty, miten funktioita soveltamalla ne jaetaan osakokonaisuuksiin. Esimerkiksi funktioon FC100 on sijoitettu solua 1 ohjaava kokonaisuus ja FC101 ohjaa solua 2. (Automaatiotekniikka/Timo Suvela 2011.)

6 VERSIONHALLINTAJÄRJESTELMÄ

6.1 Versionhallinta

Versionhallinta on järjestelmä, joka seuraa tiedostoa luontivaiheesta tiedoston poistamiseen. Versionhallinnalla voidaan palata aina aikaisempaan tiedostoon ja tutkia mitä tiedostolle on tehty, milloin on tehty, kuka on tehnyt, mitä on tehnyt ja alkuperäistä tiedostoa voidaan myös käyttää pohjana uuteen projektiin. (git-scm www-sivut 2018.)



Kuva 6.1 (https://git-Scm.com/book/fi/v1/Alkusanat-Versionhallinnasta)

Versionhallintajärjestelmä tekee jokaisesta dokumentista muokkauksen jälkeen oman version. Esimerkkinä versionhallintajärjestelmiä käytetään ohjelmisto tuotannossa. Ohjelmoija avaa uusimman version ja jatkaa siitä mihin on viimeksi jäänyt. Jos ohjelmoijan koodi ei jostain syytä enää toimi muutosten jälkeen, voi ohjelmoija versionhal-

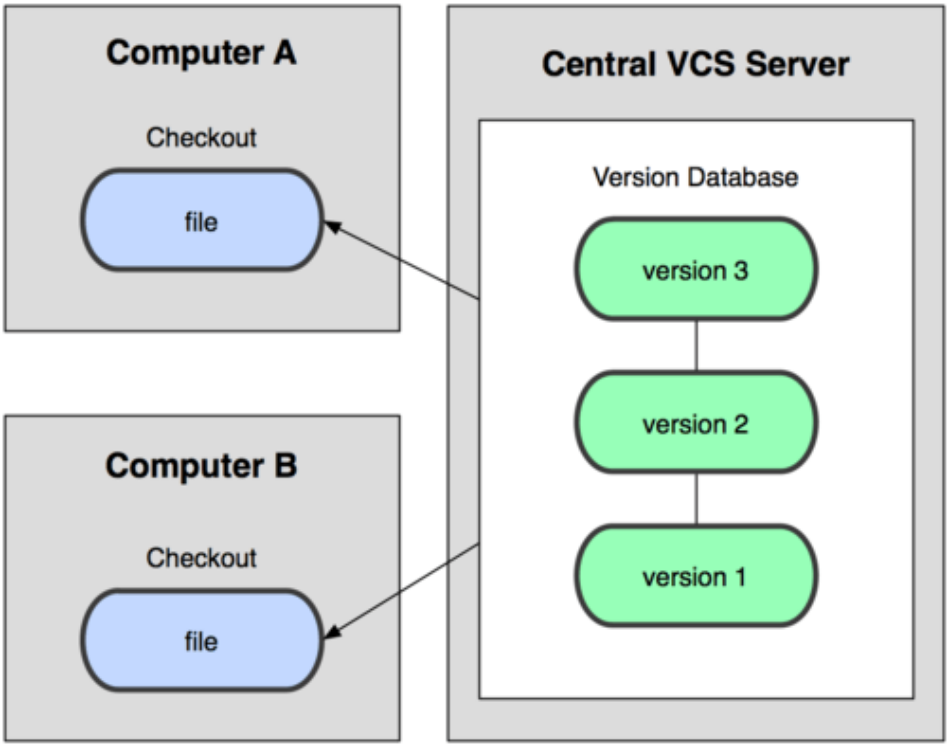
lintaa hyödyntäen avata edellisen version koodistaan ja verrata nykyiseen mikä mahdollisesti meni vikaan tai vaikka aloittaa uudestaan aiemmasta versiosta. (git-scm www-sivut 2018.)

Ohjelmoija voi tehdä lähdekoodin ja käyttää sitä monissa muissa ohjelmissaan. Esimerkiksi ohjelmoija kirjoittaa tietyntyyppisen ohjelman, jolla ohjataan kuljetinta. Tätä ohjelmaa voidaan soveltaa tai ladata sellaisenaan uusiin kuljettimiin. (git-scm www-sivut 2018.)

Versionhallintajärjestelmällä nopeutetaan ja helpotetaan ohjelmistojen ja lähdekoodien tekemistä ja uudelleen käytettävyyttä. Versionhallintaa voi tehdä manuaalisesti eli paikallisesti, mutta se on liian riskialtista, koska on mahdollista vahingossa tallentaa väärän version päälle tai väärään kansioon. Siksi yrityksillä on omien käyttötarkoituksen mukaiset versionhallintajärjestelmät. (git-scm www-sivut 2018.)

6.2 Keskitetty versionhallintajärjestelmä

Keskitetyllä versionhallinnalla tiedostot ovat kaikkien saatavilla, mutta tallennus tapahtuu yleensä tietylle serverille. Projekteja seurataan keskitetyllä järjestelmällä, koska versioiden määrää ja kommentteja voidaan tarkastella sitä mukaa kun niitä tulee. (git-scm www-sivut 2018.)



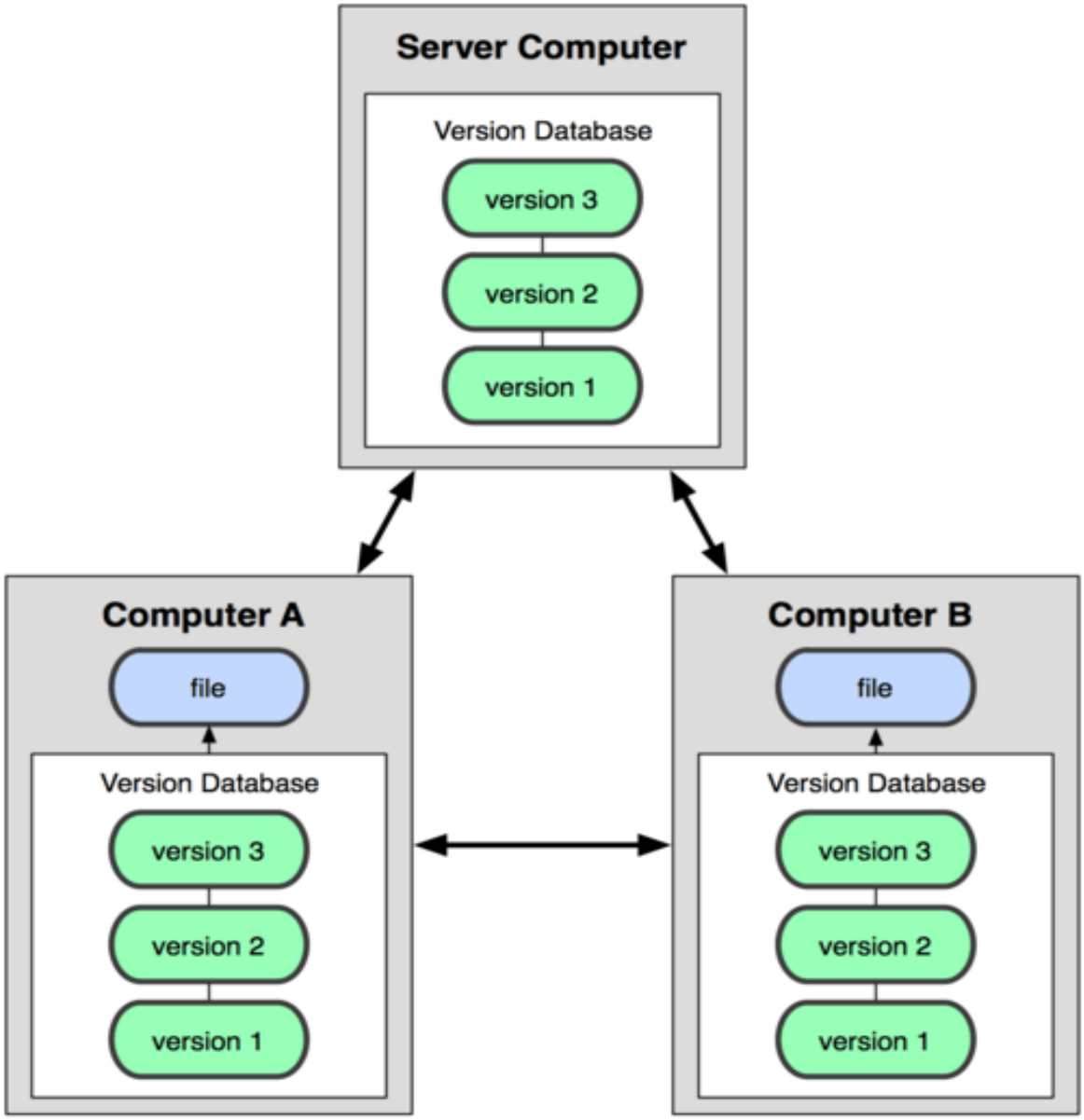
Kuva 6.2 Cenral VCS Server on palvelinkone, johon ollaan yhteydessä muilta koneilta. (https://git-scm.com/book/fi/v1/Alkusanat-Versionhallinnasta)

Keskitetyllä versionhallintajärjestelmällä on heikkoutensa. Jos keskitetty paikka menettää yhteyden tai sähkökatkon aikana tapahtuu jotain vahinkoa tiedostoille tai ylipäätään palvelinkoneelle, johon tietoa tallentuu, on mahdotonta päästä käsiksi projekteihin (Kuva 5.2). Keksitetty järjestelmä on edelleen yleinen tapa tallentaa ja seurata tiedostoja. (git-scm www-sivut 2018.)

6.3 Hajautettu versionhallintajärjestelmä

Hajautetulla versionjärjestelmällä ei ole pelkoa tiedostojen menetyksestä, koska etäkoneita ja serverikoneita on useampi. Koneet toimivat yhteistyössä toistensa kanssa reaaliajassa ja peilaavat toistensa tiedostot jatkuvassa syklissä. Jos jokin koneista menettää yhteyden toisiin koneisiin, voidaan mitä tahansa muuta konetta käyttää ja palauttaa toiseen koneeseen menetetyt tiedostot heti koneen tulessa taas käyttöön. (git-scm www-sivut 2018.)

Hajautettu versionjärjestelmä on kannattavin ja paras vaihtoehto dokumenttien ja muiden tiedostojen säilyvyyden kannalta. Enää ei tarvitse kuin luottaa versionhallintajärjestelmän työkaluihin tiedostojen päivityksen yhteydessä. (git-scm www-sivut 2018.)



Kuva 6.3 (https://git-scm.com/book/fi/v1/Alkusanat-Versionhallinnasta)

Esimerkiksi ohjelmoija lähtee toimistolta työmaalle, jolloin projekti on vain hänellä ja tässä tapauksessa hän ei ole myöskään yhteydessä muihinkaan koneisiin tai servereihin. Ohjelmoija tekee uuden ohjelman tai muokkaa vanhaa ja tallentaa omalle koneelle (Computer A). Kun ohjelmoija palaa toimistolle ja saa yhteyden muihin koneisiin ja serverikoneelle voi hän nyt päivittää versionhallinnan avulla muille koneille oman uuden version projektistaan. (Kuva 5.3)

7 VERSIONHALLINTAJÄRJESTELMIÄ PLC LÄHDEKOODEILLE

7.1 Johdantoa

Sain tutkittavakseni eri valmistajien ohjelmistoja, joita harkitaan Cimcorpin käyttöön versionhallinnan ja ohjelmiston ominaisuuksien kannalta. Lähdin tutkimaan Cimcorpin listaamia valmistajia: Bosch Rexroth, Rockwell, Siemens ja Novotekin ohjelmistoja. Cimcoropilla käytetään kyseisien valmistajien laitteita ja ohjelmistoja, mutta Cimcorp haluasi yhden ohjelmiston, jolla voisi versionhallita kaikkien ohjelmistojen tiedostoja ja mieluiten samanaikaisesti usean eri käyttäjän toimesta.

7.2 Bosch Rexroth

Indraworks on Bosch Rexrothin ohjelmistotyökalu, jolla voi suunnitella, ohjelmoida, käyttöönottaa ja versionhallita. Se on myös laaja ohjelmisto, joka mahdollistaa suunnittelun, ohjelmoinnin, käyttöönoton ja eri sovellusten diagnosoinnin koko elinkaaren ajan. Indraworks ohjelmistot ovat kaikki erikseen asennettavissa eli on hajautettu ohjelmisto, joilla jokaisella on omat toimintansa. (Bosch Rexroth www-sivut 2019.)

Software tool IndraWorks Engineering on Bosch Rexrothin mukaan sellainen työkalu, jota käytetään suunnittelussa. Siinä on ohjelmointiympäristö kaikille PLC-toiminnoille ja laajat ohjelmistokirjastot. IndraWorks Engineering:llä voi konfiguroida, asettaa parametreja, ohjelmoida ja saada diagnostiikan. Ohjelmistoa käytetään Bosch Rexroth:n käyttö- ja ohjausjärjestelmien kanssa. (Bosch Rexroth www-sivut 2019.)

IndraWorksin avulla voidaan käsitellä PLC-pohjaisia automaatio- ja ohjaustoimintoja omassa selkeässä ohjelmistoympäristössä. Ohjelmistossa on kaikki tarvittavat perustyökalut. IndraWorks sisältää tutut ohjelmoinnin komponentit, jotka ovat IEC 61131-3 standardin mukaiset. (Bosch Rexroth www-sivut 2019.)

Versionhallinta toimii keskitetysti ja omaa ominaisuuden ”Hijack”, jolla voi ottaa itselleen projektin, joka on jo muokattavana. Tämän kanssa täytyy olla varovainen, ettei

palauta toisen jo muokkauksessa olleen projektin päälle. Versioinnissa saadaan selville viimeisimmät versiot ja varmuuskopiot. Ohjelmisto kertoo myös, kuka on tehnyt muutoksia, milloin on tehty, mitä on tehty ja miksi. Indraworksin versionhallinnassa pystyy vertailemaan myös toisesta projektista oman projektin ohjelmointia. (Bosch Rexroth www-sivut 2019.)

7.3 Rockwell Automation / Allen-Bradley

FactoryTalk AssetCentre on Rockwell Automationin oma työkalu automaation liittyvien tietojen suojaamiseen, hallintaan, versiointiin, seurantaan ja rapotointiin. Tällä työkalulla täytyy olla pääkäyttäjiä tai ylimmän johdon valvontaa toimiakseen tarkoitetulla tavalla. FactoryTalk AssetCentren avulla voi valvoa ja turvata pääsyä järjestelmään, seurata version tapahtumia. Ohjelmistossa on myös automaattinen varmuuskopiointi. (Rockwell Automation www-sivut 2019.)

Ohjelmiston ominaisuuksiin kuuluu latausmahdollisuus eri manuaaleihin ja uusimpiin päivityksiin, heidän omilta web-sivuiltaan ja keskitetyn tietokannan automaattisen varmuuskopioinnin helpottamiseksi. Ohjelmisto kerää käyttäjien tiedot käyttäjäkohtaisesti järjestelmän käytön helpottamiseksi. (Rockwell Automation www-sivut 2019.)

Ohjelmiston etuja ovat suojattu pääsy ohjausjärjestelmään, joka estää ei-toivottuja muutoksia käynnissä olevissa prosesseissa tai tiedostoissa. FactoryTalk AssetCentre:llä versionhallintajärjestelmällä tiedetään aina, kuka on tehnyt, mitä on tehnyt, koska on tehnyt ja mikä on viimeisin tiedosto versio. (Rockwell Automation www-sivut 2019.)

7.4 Siemens

Siemensillä ei ole omaa versionhallintajärjestelmää, tai ei ainakaan kaupallista ohjelmistoa, joka toimisi muiden kuin Siemensin ohjelmistojen kanssa. Siemens suosittelee asiakkailleen versionhallintajärjestelmiä, kuten Versiondog ja MDT-AutoSave. (MDT-Software www-sivut 2019.)

MDT-AutoSave on versonhallintajärjestelmä ja MDT Software sen luoja, joka on vuodesta 1987 lähtien ollut katastrofien elvytys- ja muutosratkaisujen toimittaja. Yritys keskittyy pelkästään teollisuusmarkkinoiden muutosten hallinnan ohjelmistoratkaisuihin. (MDT-Software www-sivut 2019.)

MDT Software lupaa MDT-AutoSaven toimivan saumattomasti erilaisten editoripakettien kanssa. AutoSavella voi hallita versioita, arkoistoida ja varmuuskopioida tiedostoja. MDT Software toimii PLC, CNC, Robottien, documenttien ja muiden komponenttien kanssa. AutoSave toimii verkon ulkopuolella voidakseen toimia työmaalla, oman toimipisteen ulkopuolella. Näin analysoinnin ja päivitetyn version voi päivittää myöhemmin serverille omalta toimipisteeltä. (MDT-Software www-sivut 2019.)

AutoSaven versioinnilla voi tarkastella historiatietoja laitekohtaisesti ja perua muutoksia yksinkertaisesti. Ohjelmistolla pystyy myös esimerkiksi vertailemaan keskusyksikössä olevaa ohjelmaa ohjelmoijan versioon suoraan tietokoneen ja laitteen välillä. (MDT-Software www-sivut 2019.)

7.5 Versiondog

Novotek on Ruotsalainen yritys, joka toimittaa IT- ja automaatioratkaisuja. Pääpainopisteenä valmistavan teollisuuden tuotanto. Novotek on maahantuoja Versiondog-automaatiojärjestelmien version- ja konfigutaationhallinta järjestelmälle. AUVESY on saksalainen yritys, joka kehitti Versiondogin vuonna 2007, ja sitä on toimitettu tähän mennessä 30 maahan. (Novotek www-sivut 2019.)

Versiondog-ohjelmistolla on tapa yhdistää ja verrata ohjelmakoodia, parametreja ja muita tietoja. SmartCompare-vertailuominnallisuus on Versiondog:in ohjelma, mikä sisältää myös valmiit laiteliitynnät yleisimpiin laitevalmistajien automaatiolaitteisiin. Versiondog-ohjelmisto tukee robotti-, liikkeenhallintalaitteita. (Novotek www-sivut 2019.)

Versiondogilla saadaan selville automaatiolaitteiden viimeisimmät versiot ja varmuuskopiot. Ohjelmisto kertoo myös, kuka on tehnyt muutoksia, milloin on tehty, mitä on tehty ja miksi. Myös offline-muutoksista versiondog osaa verrata erilaisuudet. (Novotek www-sivut 2019.)

Versiondogin ominaisuuksiin kuuluu integroitu dokumentointi, jossa jokaisen dokumentin tiedot ovat järjestelty selkeästi ja ovat helposti löydettävissä, kuten täydellinen muutoshistoria. Versionhallinta on standardisoitu ohjaus ohjelmistojen muutosten hallintaa varten. Versiondogin versionhallintaohjelmistolla on monia vertailumahdollisuuksia kuten graafinen, taulukko tai tekstimuodossa oleva muutosten vertailumahdollisuus. Ohjelmistosta on haluttu, että muutosten tarkastelu olisi selkeää ja ymmärrettävä. (Novotek www-sivut 2019.)

Versiondogilla on myös automaattinen varmuuskopiointimahdollisuus. Varmuuskopioinnin ominaisuuksia ovat täysin turvatut tiedot ja konfiguroiva hälytys, kun ohjelmisto havaitsee eroja. Versiondog omaa säännölliset ja automaattiset online-offline-vertailut. Myös Mobile BackupClient mahdollistaa saman varmuuskopiointistrategian käytön sekä verkossa, että ei-verkossa olevissa laitteissa. (Novotek www-sivut 2019.)

7.6 Versionhallintajärjestelmien vertailutaulukko

	Versiondog	Bosch Rexroth	Rockwell Automation	MDT-Software
Tukee eri ohjelmointi ympäristöjä	X	-	-	-
Oma ohjelmointi ympäristö	-	X	X	X
Näyttää version historian	X	X	X	X
Muutosten vertailu	X	X	X	X
Useita käyttäjiä, sama projekti	-	-	-	-
Automaattinen varmuuskopiointi	X	-	X	X

Tein vertailutaulukon versionhallintajärjestelmistä, jotta nähdään kaikkien olevan ominaisuuksiltaan lähes samanlaiset. Versiondog vie voiton, koska se tukee ohjelmointiympäristöjä, kun taas muut toimivat omilla ohjelmistoillaan ja jotkut vain joidenkin tiettyjen valmistajien kanssa. Missään versionhallintajärjestelmässä ei ole mahdollista, että yksi projekti saataisiin useammalle ohjelmoijalle. Projekti kokonaisuudessaan tulee vain yhdelle versioitavaksi. Esimerkiksi monen kuljettimen ohjelmointia ei saa yksittäin jaoteltua kuljetinkohtaisesti eri henkilölle.

8 VERSIONHALLINTAJÄRJESTELMÄN VALINTA JA TUTKIMUS

8.1 Novotek, Versiondog testaukseen

Cimcorp on järjestelmätomittaja ja työskentelee eri valmistajien kanssa yhteistyössä, ja kun Versiondog pystyy ”keskustelemaan” monen eri laitevalmistajan ja ohjelmiston kanssa, niin valitsimme Versiondog:n tarkempaan testiin. (Novotek www-sivut 2019.)

Versiondog on AUVESY:n luoma versionhallintajärjetelmä, ja sen toiminta perustuu ominaisuuteensa yhdistää ja verrata koodia ja parametreja.
Versiondog sisältää yleisimpiin automaatiolaitteisiin laiteliitynnät, jotta ei tarvita laitevalmistajien omia ohjelmistoja, mikä mahdollistaa tietojen vertaamisen aiemmin tallennettuihin tietoihin. Versiondog toimii siis omana versionhallintaohjelmistona ja toimii yhteistyössä muiden ohjelmistojen kanssa. (Novotek www-sivut 2019.)

Lähdin tiedustelemaan maahantuojalta, Novotekilta, olisiko mahdollista saada Versiondog:sta demo-versiota tietokoneelle asennetussa muodossa. Cimcorp ei halua mielellään käyttää web-pohjaisia versioita. Samalla kysyin, voisiko asiantuntija tulla pitämään meille esittelyn Versiondog-ohjelmistosta ja kertomaan mitä ohjelmistolla pystytään tekemään. Siten saataisiin hyvät eväät ja varsinkin faktaa siihen, että mitä Versiondog:lla voidaan ylipäätään tehdä ja mitä voidaan lähteä testaamaan.

8.2 Novotekiltä Versiondog

Novotekiltä vastattiin, että heillä on tapana hoitaa asioita pilottihankkeena ja demon järkkääminenkin onnistuu. Tämän pilottihankkeen ideana on, että Novotek auttaa asiakasta systeemin pystyttämisessä pienemmässä mittakaavassa. Siinä asiakas, tässä tapauksessa Cimcorp, saa koulutuksen sekä samalla voidaan kartoittaa kaikki tarpeet, jonka perusteella Novotek voi tarjota lopullista ratkaisua versionhallintaan. (Novotek www-sivut 2019.)

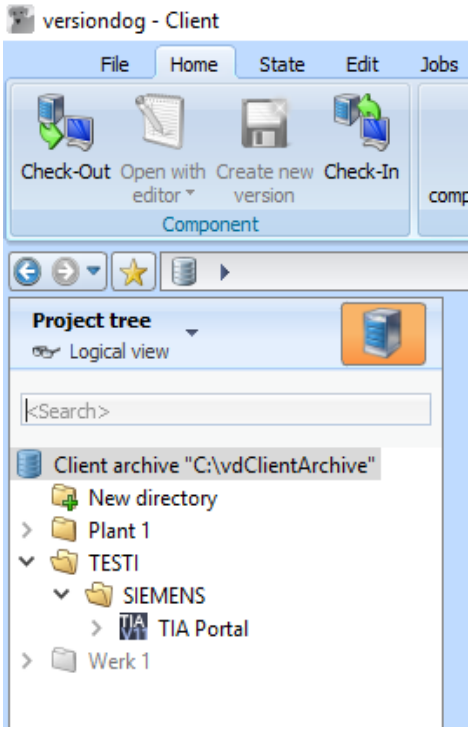
Sovimme koulutuksen ajankohdan, ja Novotekilta tulee kaksi henkilöä esittelemään ohjelmistoa. Toinen kertoo yleisesti, mikä Versiondog on ja toinen (ohjelmistoasiantuntija) kertoo ohjelmiston käytöstä.

8.3 Versiondog esittely

Timo Porkola tuli esittelemään Versiondogia Cimcorpille. Hän kertoi, mitä Versiondog:lla voi tehdä ja kuinka moni maailmalla käyttää ohjelmistoa. Porkolan esityksestä selvisi, että Versiondog-ohjelmisto on varmuuskopioinnin ja versioinnin työkalu. Ohjelmisto editoreita käytetään niin kuin ennenkin, mikä tarkoittaa, että Versiondog toimii omana versionhallintaohjelmistonaan. Esimerkiksi PLC-koodi voidaan asettaa varmuuskopioitavaksi, milloin ikinä haluaa, koodin muuttuessa automaattiset varmuuskopioinnit ovat varma tapa pelastaa tiedostot. Versiondogia mainostettiin myös helppona käyttää, kuten Porkola sanoi: ”Perustoiminnallisuus neljän askeleen takana”.

Ohjelmistoa tuli esittelemään Mikko Kovalainen, joka näytti Versiondog:n toiminallisuutta omalta tietokoneeltaan. Kovalainen näytti askel askeleelta, miten neljän vaiheen käyttö onnistui. Neljävaihetta on kuvan 8.1 yläkulmassa: “Chek-Out”, “Open with editor”, “Create new version” ja “Check- In”. (Kuva 8.1). ”Check-Out” ottaa projektin itselle muokattavaksi, jonka jälkeen avataan ohjelmistoympäristö ”Open with editor”. Projektia voi tässä kohtaa muokata normaalisti. Kun muokkaukset ovat valmiit, tallennetaan ohjelmistoympäristö ja palataan versiondogiin ja luodaan uusiversio ”Create new version”. Tämän jälkeen kommentoidaan ja vertaillaan muutoksia, jonka jälkeen palautetaan projekti serverille uutena versiona ”Check-In”.

Esimerkiksi kun aloitetaan uutta projektia, niin ensimmäisenä luodaan Projekti kansiot ja komponentti Versiondog:lle, jonka jälkeen avataan ohjelmaeditori, millä tehdään muokkaukset komponentille. Komponentti on tässä esimerkissä Siemensin TIA Portal projekti (Kuva 8.1).



Kuva 8.1 (Versiondog - Client Versio 6.5)

Muokkaukset tehdään normaalisti Siemens TIA Portal ympäristössä, ja kun muokkaukset on tehnyt, voidaan tallentaa ja palata taikaisin Verisondog:lle. Kun ollaan luotu komponentti eli projekti, mikä voi olla esimerkiksi Siemensin, voidaan tämän jälkeen tallentaa se Versiondog serverille ja Versiondog luo automaattisesti version järjestelmään. Tämän jälkeen kommentoidaan, mitä kyseiselle komponentille on tehty, jotta tiedetään, missä vaiheessa projekti etenee.

Yleisesti teollisuudessa, joku voi muokata kerran tai kaksi kuukaudessa koodia ja tallentaa ja versioida Versiondog:n avulla, mutta Cimcorp:n tapauksessa laitteita valmistellaan ja ohjelmoidaan Cimcorp:n useiden insinöörien toimesta ja vielä samalla projektilla. Saman projektin muokkaaminen vielä vaikuttaisi olevan hankalaa Versiondog:lla, mutta selvästi helpottaisi tämän hetkistä tilannetta, kun voidaan hallita järjestelmiä yhden ohjelmiston avulla.

8.4 Versiondog demo

Versiondog testauksessa testasin Siemens, Rockwell ja Bosch Rexrothin ohjelmistojen versiointia versiondogilla. Pyysin käyttööni vanhoja projekteja, jotka tallensin omalle koneelle ja sitä kautta versiondogille.

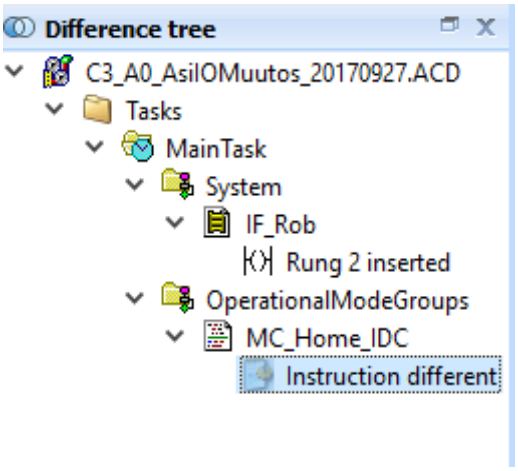
Testasin ensin Siemensin TIA Portal ohjelmointiympäristön avulla versiondogia. Loin Versindogiin kansion TESTI ja alikansioksi SIEMENS, jotta pysyisin omassa testailussa mukana, että mikä projekti missäkin sijaitsee. Muuta tarkoitusta kansioilla ja alikansioilla ei tässä kohtaa ole, ja oman kansiohierarkian saa varmasti jokainen yritys tehdä omalla tavallaan. Loin kansioon SIEMENS ”komponentin”. Eli listalta etsin Siemens, ja sitä kautta TIA Portal ja loin komponentin (Kuva 8.1). Näin määritin millä ohjelmistoympäristöllä kyseinen projekti avataan myöhemmässä vaiheessa muokkaukseen, tässä tapauksessa Siemens TIA Portal. Seuraavaksi jouduin etsimään TIA Portalin projektikansioista projektin ja ”raahasin” sen versiondog:n komponentin päälle, jonka olin juuri luonut. Näin sain Versiondog serverille Siemens projektin ja olin valmis testaamaan, miten uuden version luominen onnistuu.

Itse versiondogin käyttö tuntui helpolta ja siihen pääsi aika äkkiä kiinni, ja tietenkin sillä voi tehdä paljon enemmän, mitä testivaiheessa ei tajunnut edes kokeilla. Pääasia oli kuitenkin testata, kuinka versiointi onnistuu eri ohjelmistoilla. Testasin siis Siemenssin ohjelmistoa ensin ja avasin ”editorin”. Kuten aikaisemmin olin määrittänyt, versiondog avasi Siemens TIA Portal -ohjelman ja samaisen projektin, jonka olin kohdekansioon luonut.

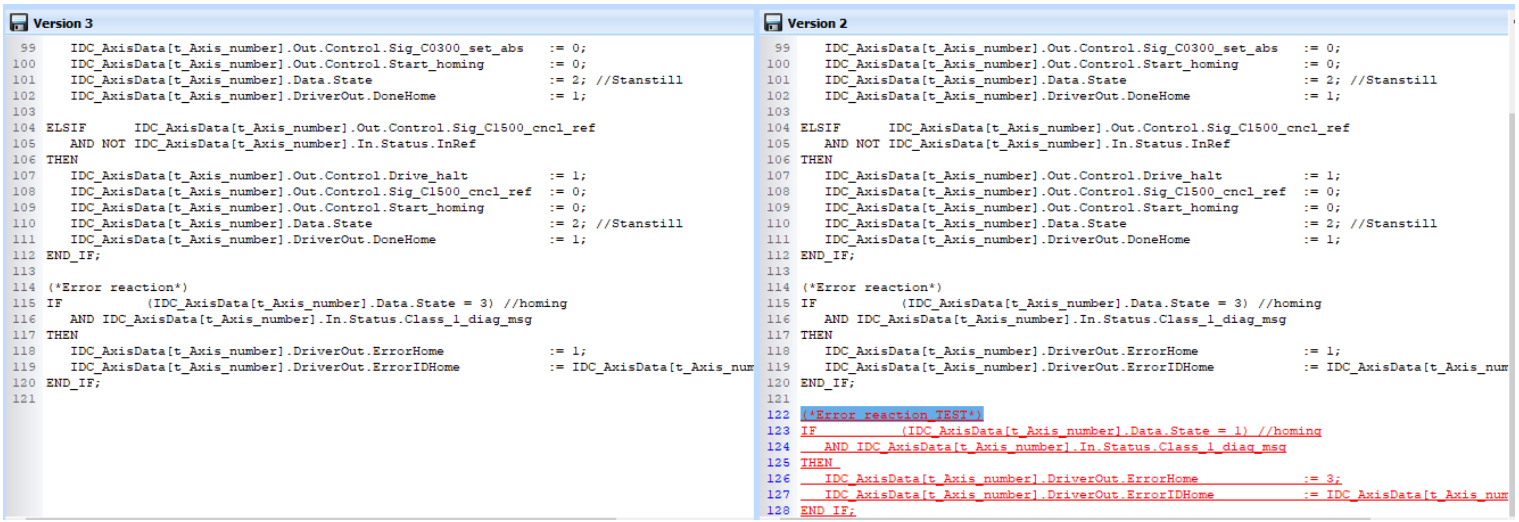
Lähdin muokkamaan ST (Structure Text) muodossa olevaa ohjelmointitekstiä. Lisäsin vain yksinkertaisen käskyn, koska ohjelman toimivuudella ei tässä tapauksessa ollut merkitystä. Tein saman Ladder Diagram ohjelmointikielellä ja tallensin TIA Portalin ja suljin ohjelmiston. Tämän jälkeen loin uuden version TIA Portal -projektista ja samalla automaattisesti versiondogin vertailuohjelma aukesi eteeni. Siinä ne muutokset olivat, jotka olin tehnyt, ja joissa kansiossa olin tehnyt. (Kuva 8.3) Todella selkeää ja helppoa.

Ainoa asia, joka ihmetyttää tässä testauksessa, on se, kun muokkasin projekteja uudestaan, niin versiondog ei muistuta missään kohtaa, että pitäisi kertoa johonkin, mitä muutoksia tein. Sellainen kohta on kyllä olemassa, johon kirjoitetaan mitä kukakin on tehnyt ohjelmaan, mutta itse monesti unohdin kokonaan tämän.

Testasin samalla tavalla Rockwellin Logix Designer ohjelmiston ja muokkasin olemassa olevaa ohjelmaa kuten Siemensillä ja versiointi toimi kuten siemesillä.



Kuva 8.2 Versiondogin muutospuu tulee editorin suljettua näkyviin ja voit tarkastella kaikkia muutoksiasi ennen uuden version luontia. (Versiondog, Versio 6.5 2019.)



Kuva 8.3 (Vertailu näyttää tältä, vain ohjelmointikieli muuttuu. Punaisella värillä on huomioitu ohjelman muutos. (Versiondog, Versio 6.5 2019.)

Testasin viimeisenä Bosch Rexrothin ja tämä ei toiminut ollenkaan niin luontevasti kuin aikaisemmat testit. Bosch Rexrothin Indraworks ohjelmisto ei auennut ollenkaan editoria aukaistaessa. Kävi ilmi, että kaikkiin ohjelmistoeditoreihin ei ole varauduttu vaan oletus asetuksiin on luotu jonkin yleisimmistä ohjelmistoeditoreista ja Cimcorpilla kun on eri versio Indraworksistä, niin oletus editori ei voinut aueta. Asetuksista piti lisätä projektitiedoston tyyppi, jotta editoria avatessa versiondog osaisi avata projektin.

Versiondog toimii lisenssillä ja lisenssin uusiminenkin oli helppoa. Pyysin lisää aikaa testaukseen ja sain uuden kuukauden mittaisen lisenssin. Ohjeet tulevat automaattisesti lisenssin mukana sähköpostiin. Lisenssi ladataan koneelle ja avataan versiondog adminclient, ja sieltä päivitetään lisenssi muutaman klikkauksen jälkeen. Ohjelma myös ilmoittaa onnistuiko lisenssin päivitys.

9 YHTEENVETO

Cimcorp Oy:n toiveena oli saada olemassa olevista versionhallintajärjestelmistä itselleen omien tarpeiden mukainen ohjelmisto ja käyttää sitä PLC lähdekoodeille. Versiondog oli tutkimusten ja testien perusteella paras ohjelmisto, koska Cimcorp Oy on järjestelmätoimittaja ja luodessaan järjestelmiä, heillä on kentällä useampi automaatiovastaava. Joten tutkittiin ohjelmistoja, joissa olisi mahdollisuus versioida ja työstää projektia saman aikaisesti. Tämä osoittautui mahdottomaksi jokaisessa versionhallintajärjestelmässä. Ominaisuuksien osalta ohjelmistot olivat lähes samanlaiset. Versiondog puolestaan tukee Cimcorp Oy:n käyttämiä ohjelmistoja, minkä takia versiondog on ylivoimaisesti paras ratkaisu. Ilmeni muutamia ongelmia kuten tietyn ohjelmistoympäristön aukaiseminen, mutta ratkaisu löytyi siihenkin pienellä avustuksella ja asetuksia muuttamalla.

Tehtävänäni oli myös laatia versionhallintajärjestelmälle selkeä ohjeistus, joka kattaa yksinkertaiset ohjeet käyttöä varten, mikä tulee olemaan pohjana Cimcorp Oy:n käytössä ja vain asiakkaan versioissa.

Mielestäni projekti oli sopivan haastava ja mukava toteuttaa. Pääsin tutustumaan syvemmin automaation versioinnin saloihin ja ongelmiin. Sain testata eri ohjelmistoja, joita koulussa ei olla testattu. Sain myös kokea epäonnistumisia ja onnistumisen iloja.

LÄHTEET

www.plcopen.org/pages/tc1_standards/ , TC1 –Standards. (Vitattu 27.09.2018)

www.plctutor.com/ , (Viitattu 27.09.2018)

www.cnc-tekniikka.com/CNC-forum1/index.php?topic=6307.0 , (PLC-ohjelmointi foorumi) Viitattu 03.11.2013

www.techopedia.com/definition/547/source-code , Viitattu 01.10.2018

www.sesko.fi/files/101/osio_9.pdf , (Logiikka ja standardit) Viitattu 4.10.2018

https://git-scm.com/book/fi/v1/Alkusanat-Versionhallinnasta , Viitattu useasti kappaleessa 6 versionhallintajärjestelmä 04.10.2018

https://www.novotek.com/fi/novotek#dsection , Viitattu 19.10.2018

https://www.rockwellautomation.com/ , Viitatu 30.10.2018

http://li-pas.uwasa.fi/~TAU/AUTO1060/slides.php?Mode=Printer&File=9050PLC.txt&MicroExam=On 06.11.2018 (Tikapuukieli)

http://dailyautomation.sk/01-programovacie-jazyky-plc/ 06.11.2018 (FBD)

https://www.quora.com/What-is-the-best-programming-language-to-program-PLCs 06.11.2018 (ST-kieli)

http://www.mdt-software.com/autosave/ Viitattu 06.11.2018

https://www.plcacademy.com/ Viitattu 10.01.2019, Viitattu useaan kertaan kappaleessa 3 (Ohjelmointikielet).

https://www.motioncontroltips.com/ Viitattu 15.01.2019, Viitattu useasti kappaleessa 3 (Ohelmointikielet)

Cimcorp Oy intranet, Viitattu useaan kertaan kappaleessa kaksi (YRITYS).

Satakunnanammattikorkeakoulu opetusmateriaali Timo Suvela (2010), viitattu useasti kappaleessa 5 (PLC- Ohjelmointi.)

LIITE 1

Versiondog asennus- ja käyttö. 18 sivua. Liite on Cimcorp Oy:n sisäinen ja luottamuksellinen dokumentti ja saatavilla vain asiakkaan versiossa.