Juho Keskinen

KÄYTTÖTAPARYHMÄN OHJELMOINNIN AUTOMATISOIMINEN ROCKWELLIN YMPÄRISTÖSSÄ

Sähkö- ja automaatiotekniikan koulutusohjelma

2019

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

KÄYTTÖTAPATYHMÄN OHJELMOINNIN AUTOMATISOIMINEN ROCKWELLIN YMPÄRISTÖSSÄ

Keskinen, Juho
Satakunnan ammattikorkeakoulu
Sähkö- ja automaatiotekniikan koulutusohjelma
Syyskuu 2019
Sivumäärä: 42
Liitteitä: 2

Asiasanat: käyttötaparyhmä, automaatio, Rockwell, Programmable logic controller, Application code manager

---

Käyttötaparyhmän ohjelmoinnin automatisointi vähentää yksitoikkoisen manuaalisen työn määrää ja se voidaan automatisoida.

Automatisoinnin mahdollisuuksia tutkittiin Rockwellin ohjelmistoympäristössä ja Application code managerin (ACM) toiminallisuutta käyttötaparyhmän luonnissa. Selvitettiin kahden eri toimintatavan toimivuutta ja päädyttiin tapaan 2.

Tavan 1 mukainen käyttötaparyhmän luonti olisi toteutettu täysin ACM:n sisällä käyttäen hyväksi kirjasto-objektien parametrointia ja massakopiointia. ACM:n nykyinen versio (3.10.00) ei kyennyt riittävän dynaamiseen ohjelmanluontiin vaan kirjasto-objektien käsittelymenetelmä esti Cimcorpin käyttämän ohjelmistorakenteen luonnin ACM:n sisällä.

Päädyttiin tapaan 2, jossa ACM toimii standardimoduulien kirjastona ja Excelissä oleva makro hallinnoi ja luo halutut käyttötaparyhmät L5X-tiedostoksi, joka voidaan viedä Logix designeriin haluttuun projektiin. Excel-lähdetiedoston toimiessa tietokantana työkalulle.

Työkalun käyttöä varten luotiin graafinen käyttöliittymä, jonka välityksellä loppukäyttäjä generoi käyttötaparyhmien PLC-koodit (Programmable logic controller) L5X-tiedostoon. Käyttöliittymä integroitiin Excel-lähdetiedostoon, jolloin työkalu ja lähdetiedosto kulkevat yhtenä pakettina, helpottaen sen jakamista Cimcorpin sisällä ja tulevaisuuden muokkaamista.

AUTOMATING OPERATION MODE GROUP PROGRAMMING IN ROCKWELL'S APPLICATIONS

Keskinen, Juho
Satakunta University of Applied Sciences
Degree Programme in Electrical and automation engineering
September 2019
Number of pages: 42
Appendices: 2

Keywords: Operation mode group, automation, Rockwell, Programmable logic controller, Application code manager

---

Operation mode groups (OMG) programming is a laborious work phase, which can be automated thus reducing the work amount.

The possibilities of automation were studied in Rockwell's software environment and the functionality of Application code manager (ACM) in creation of operation mode groups. Two different operation strategies and their applications were researched and method 2 was selected.

In method 1, the generation of the OMGs would have been executed solely by the ACM and its functions to parameterized library objects and mass copy said library objects to create the OMGs. The current version of ACM (3.10.00) was not able to dynamically create the OMGs software structure in a way that Cimcorps software structure was maintained. This restriction made the solely use of ACM impossible.

In the selected method 2, ACM manages the library of standardized modules and a macro in Excel manages the creation of the OMGs PLC-code (Programmable logic controller) into a L5X-fileformat. The created L5X-file can then be imported to Logix designer into a selected project. The Excel-file acts as a database for the macro.

A graphical user interface was created for the macro and it's used as centralized control point for the end user to create the PLC-code into L5X-file. The GUI was integrated into the Excel-file, so the GUI, macro and database move as a single unit to simplify in-house sharing in Cimcorp and modifications to the macro in the future.

SISÄLLYS

1 JOHDANTO

Käyttötaparyhmien luonti on yksitoikkoinen työvaihe, joka on automatisoitavissa, näin tuoden nopeutta projektien läpiviemiseen. Vaihe on automatisoitavissa ja tutkinnan kohteena ovat mahdollisuudet, jotka auttavat Cimcorpia luomaan projektien PLC-ohjelmointia (Programmable logic controller) nopeuttavan työkalun. Työkalu luo annettujen parametrien perusteella käyttötaparyhmän/ryhmiä, jotka ovat välttämättömiä PLC-ohjausta luodessa. Opinnäytetyössä otetaan kantaa erilaisiin vaihtoehtoihin työkalun toteuttamiseen ja miten sen käytettävyys, toiminnallisuus ja tarkoituksenmukaisuus täyttyy sekä muodostuu. Työkalun on tarkoitus luoda Rockwell:n Studio 5000 ympäristöön PLC-koodi. Eritoten käyttäen hyväksi Application code manageria (ACM), Library designeria, Library object manageria sekä Logix designeria. ACM:n on tehty Excel yhteensopivuus, jonka avulla kirjasto-objektien massakopiointi käy nopeasti. Tätä massakopiointia tullaan tutkimaan ja sen mahdollisuuksia osana työkalua olemista selvitetään. Vaihtoehtoisena tapana tullaan tutkimaan Excel VBA-pohjaista työkalua, joka luo valmiin XML-pohjaisen L5X-tiedoston, joka voidaan viedä Logix designeriin, jolloin lopputulos on valmis PLC-ohjelmakoodi.

Opinnäytetyö ottaa kantaa molempiin toimintatapoihin ja niiden toteutettavuuteen sekä yhdistettävyyteen Cimcorpin nykyisten työkalujen kanssa.

2 VAATIMUKSET JA TAVOITTEET

2.1 Vaatimukset

Jotta työkalu ja sen sisältämä ohjelma saadaan luotua, on työkalulle ja sen käyttämille eri resursseille asetettu vaatimuksia. Cimcorpilta vaatimuksina ovat mahdollisuuksien mukaan nykyisen ohjelmistorakenteen säilyttäminen ja olemassa olevan Excel-lähdetiedoston integroiminen työkaluun.

Työkalun alustan on hallittava käyttötaparyhmän rakenteen luominen ja skaalattavuus. Työkaluun on kyettävä myös luomaan käyttöliittymä, johon voidaan tulevaisuudessa liittää muitakin toiminallisuuksia. Päävaatimuksina on valmiin työkalun muokkausmahdollisuus ja siirreltävyys.

2.2 Tavoitteet

Tavoitteena on saada aikaan ohjelmarunko, josta voidaan kehittää varsinainen työkalu. Työkalu vähentää yksitoikkoiseen työhön kulunutta aikaa huomattavasti ja työkalusta olisi suurta hyötyä Cimcorpille.

Työ on rajattu ohjelmarungon tekemiseen, siihen vaadittavaan tutkimiseen ja dokumentointiin.

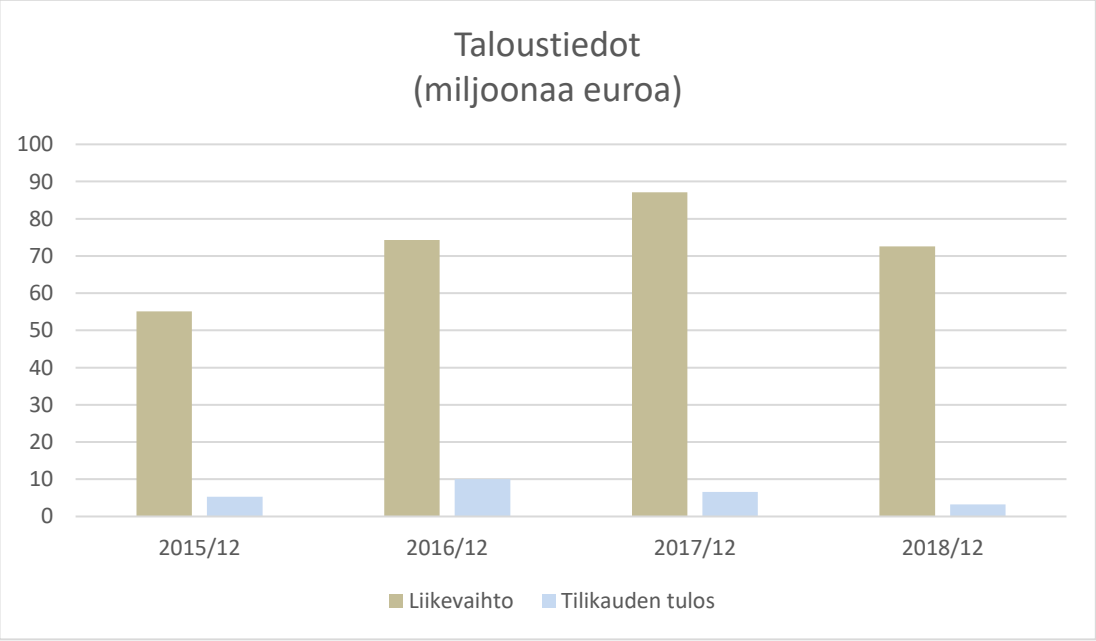
3 CIMCORP OY

Cimcorp Oy on vuonna 1975 suomessa perustettu automaatiojärjestelmien toimittaja, joka keskittyy pääasiallisesti rengas- ja elintarviketeollisuuden järjestelmäintegrointiin ja toimittamiseen. Cimcorp Group koostuu emoyhtiöstä, joka sijaitsee Ulvilassa. Konsernin tytäryhtiötä sijaitsee Kanadassa, Yhdysvalloissa ja Intiassa. Cimcorp Group on japanilaisen Murata Machineryn omistama kokonaisuus. Cimcorp Group:n kuuluu n. 300 työntekijää.

Erikoisosaamisena Cimcorp suunnittelee, ohjelmoi ja valmistaa robotiikkaan perustuvia ratkaisuja. Cimcrop myy Dream Factory-nimikkeellä teollisuusautomaatioratkaisuaan, joka toimii avaimet käteen periaatteella, jossa Cimcorp suunnittelee, integroi ja luovuttaa tehtaan tai varaston valmiina pakettina asiakkaalle. Alihankkijoiden kanssa toimiminen kuuluu myös järjestelmäintegroijan toimenkuvaan. (Cimcorp Oy 2019, www.sivut)

Cimcorpin liikevaihdosta tulee noin puolet rengasteollisuudesta ja toinen puolikas elintarviketeollisuudesta. Taulukossa 1. esitetään vuosien 2015-2018 taloustiedot. (Suomen Asiakastieto 2019, www-sivut)

Taulukko 1.Taloustiedot vuosilta 2015-2018 (Suomen Asiakastieto)



4 OHJELMISTOARKKITEHTUURI

4.1 Ohjausjärjestelmä

Ohjausjärjestelmä koostuu laitteista, jotka jaotellaan eri ryhmiin riippuen laitteiden toiminnallisuudesta ja käyttötarkoituksesta. Ryhmistä muodostuu toimialavyöhykkeitä ja jaollisuuden tarkoituksena on helpottaa ja tehdä järjestelmän valvonnasta, ylläpidosta, sekä huoltamisesta turvallisempaa. Vyöhykejako jaetaan yleisesti kolmeen osa-alueeseen: hätäpysäytysvyöhyke, turvapysäytysvyöhyke, sekä käyttötaparyhmä. Tässä työssä keskitytään enemmän käyttötaparyhmän käsittelyyn. (Suvela 2010, 3)

4.1.1 Hätäpysäytysvyöhyke

Hätäpysäytysvyöhykkeeseen kuuluu laitekokonaisuus, joilla on yhteinen hätäpysäytystoiminto. Yksittäinen laite voi kuulua useamman kuin yhden hätäpysäytysalueen piiriin ja hätäpysäytystoiminto voi vaikuttaa vain yhden hätäpysäytysvyöhykkeen alaisiin laitteisiin. (Suvela 2010, 6)

Hätäpysäytys aikaansaa vaikuttamallaan vyöhykkeellä turvapysäytyksen. Hätäpysäytyksen vapautus ja kuittaus aktivoivat hätäpysäytyspiirin, mutta turvapysäytyskuittaus vaaditaan ja turvapysäytyksen kuittaus on oltava hätäpysäytysvyöhyke kohtainen. (Suvela 2010, 6)

Hätäpysäytysvyöhykkeiden jako huomioidaan koneturvallisuuden suunnitteluperiaatteiden standardissa SFS-EN ISO 13850 kohta 4.1.2 seuraavasti:

”Hätäpysäytysvyöhyke voi kattaa osan (osia) koneesta, koko koneen tai koneryhmän. Hätäpysäytysvyöhykkeet voivat olla osittain päällekkäisiä. Hätäpysäytysvyöhykkeet on määriteltävä ottamalla huomioon seuraavat asiat:

a) Koneen fyysinen sijoittelu perustuen koneen näkyvissä oleviin alueisiin
b) Mahdollisuus tunnistaa vaaratilantiit (esim. näkyvyys, äänet, hajut)
c) Tuotantoprosessiin liittyvät turvallisuusvaikutukset
d) Ennakoitavissa oleva vaaratekijöille altistuminen

e) Mahdolliset lähistöllä olevat vaarantekijät”
(SFS-EN ISO 13850 2015, kohta 4.1.2)

4.1.2 Turvallisuusvyöhyke

”Turvallisuusvyöhykkeellä tarkoitetaan turvalaitteilla rajattua laitekokonaisuutta, johon kuuluvilla laitteilla on yhteinen turvapysäytystoiminto, jonka aktivoi myös hätäpysäytys.” (Suvela 2010, 3)

4.2 Käyttötaparyhmä

Käyttötaparyhmällä (eng. Operational Mode Group, OMG) tarkoitetaan ryhmää laitteita, jota ohjataan omana kokonaisuutenaan. Ryhmä koostuu yhdestä tai useammasta laitteesta. Käyttötaparyhmän voi sisällyttää vain yhden turvavyöhykkeen sisälle, mutta riittävän riskittömäksi perustellun käyttötaparyhmän voidaan jättää turvavyöhykkeen ulkopuolelle. Käyttötaparyhmän laitteet voivat kuulua vain yhteen käyttötaparyhmään. (Suvela 2010, 6)

Käyttötaparyhmän käyttökohteesta riippuen, ryhmät on jaettu kahteen:

I. Prosessilaitteet, jotka koostuvat itsenäisistä valittua prosessia suorittavista toimilaitteista, jotka tarvitsevat oman/omia toimitiloja, (esim. palletin purkaja, tai palletoija)
II. Kappaleenkäsittelyryhmä, mikä käsittelee kappaletavaroita, esim. laatikoita. Tähän ryhmään kuuluu esim. kuljetin, keskitin, hissi, siirto/varastorobotti ja prosessilaite, joka ei vaadi erillisiä toimitiloja. (Suvela 2010, 6)

Käyttötaparyhmät nimetään ja niille annetaan tunnukset riippuen niiden toimialueen sijainnista sekä tehtävästä. Nimeäminen tehdään yleisesti projektiin liittyvää kirjainta tai tunnusta sekä juoksevaa numeroa kertomaan käyttötaparyhmän numeron, esim. Kn, jossa n merkitsee juoksevaa numeroa. (esim. K1, K2, K3). (Suvela 2010, 6)

4.3 Käyttötaparyhmän toimitilat

Käyttötaparyhmän laitteita ohjataan yhteisellä hallintaelimellä, joka määrittelee toimitilat ryhmään kuuluville laitteille. Kun käyttötaparyhmään vaikuttava turvapysäytys ei ole päällä, voi käyttötaparyhmän omat toimitilaa hallitsevat elimet ohjata käyttötaparyhmän laitteita. (Suvela 2010, 8)

Toimitiloja on käyttötaparyhmässä aina ”SEIS”, ”KÄSIAJO” ja ”AUTOMAATTIAJO” –tilat, joita hallitaan käyttötaparyhmän yhteisessä hallintaelimessä. Nämä päätoimitilat toimivat seuraavasti:

4.3.1 Seis

”SEIS”-tilassa ollessa käyttötaparyhmään kuuluvat laitteet ovat ohjelmallisesti sellaisessa tilassa, jossa niitä ei voida ohjata. Toiminnallisesti ”SEIS”-tila keskeyttää muiden toimitilojen ohjaamat laitteet, hallitusti hidastaen pysähtyneeseen tilaan ja aiheuttaen ”KOTIINAJO”- ja ”AUTOMAATTIAJO”-sekvensseihin tilojen resetoitumisen. ”SEIS”-tila ei ole sama kuin turvapysäytystila tai sen toiminta ei aktivoi turvalaitteita automaattisesti. Poikkeuksena sellaiset järjestelmän osat, joilla ei ole turvalaitteita, on turvapysähdyksen aktivoiduttava päälle ”SEIS”-tilaan mentäessä. Näissä poikkeustapauksissa ”SEIS”-kytkin kytketään turvalaitteiden tilalle. (Suvela 2010, 9)

”SEIS”-tilan valinta tehdään, esim. näyttöpaneelilta, jossa on valintapainikkeet ”SEIS/KÄSI/AUTO”-tiloja varten. Valomerkit syttyvät indikoiden ”SEIS”-tilan aktivoitumista, tällöin ”SEIS”-painikkeen valomerkki syttyy ja kaikki muut valot, paitsi ”KOTIASEMA”-valomerkki sammuu. ”SEIS”-tila aktivoituu myös silloin kun käyttötaparyhmän turvalaitteisiin vaikutetaan millä tahansa keinolla, paitsi tuotantopysäytyksen aikana. (Suvela 2010, 9)

4.3.2 Käsiajo

Käsiajotilassa toimilaitteille mahdollistetaan käsiajo, eli toimilaitteiden pakko-ohjaukseen tarkoitettuja hallintalaitteita (painonapit, kytkimet) käyttäen ohjataan toimilaitetta manuaalisesti. Käsiajotilan ero automaattitilaan on selkeä nopeuden tiputus. Yleisesti käsiajossa käytetään nopeutta 0,25m/s. Vaikka käsiajonopeudelle on yleinen määre, tulee se asettaa aina tapaus- ja laitekohtaisesti, jotta saavutetaan turvallinen käsittelynopeus. (Suvela 2010, 9)

Käsiajotilaan päästään silloin, kun hätäpysäytys ja turvapysäytys ovat kuitattuna ja valintakytkin tai painike on asetettu ”KÄSI”-tilaan eikä ”KOTIASEMA” painike ole painettuna. (Suvela 2010, 9)

4.3.3 Automaattiajo

Automaattiajolla tarkoitettaan käyttötaparyhmän normaalia ajotilaa, jossa toimilaitteet suorittavat työkiertojaan. Automaattiajo jakaantuu kahteen eri osaan, automaattiajoon ja automaattiajon pysäytystiloihin. (Suvela 2010, 10)

Käynnistysehtoja automaattiajolle on hätä- ja turvapysäytysten kuittaus, käyttötaparyhmän oleminen ”KOTIASEMA”-tilassa ja käynnistyssignaali hallintapaneelista tai näytöstä.

4.4 Muut toimitilat

”SEIS”, ”KÄSIAJO” ja ”AUTOMAATTIAJO”-tilojen lisäksi käyttötaparyhmän hallintaan liittyy muitakin alatiloja, joiden avulla käyttötaparyhmän toimintoja hallintaan. Eri tiloja on:

- Jatkuva automaattiajo
- Tuotantopysäytys
- Tauko
- Työkierron lopetus
- Ulkoinen pysäytys
- Askellusajo

Muista käyttötaparyhmän ajotiloista löytyy tietoa esimerkiksi Satakunnan ammattikorkeakoulun kursseilta: Automaatiotekniikan perusteet ja Automaatiotekniikka. (Suvela 2010, 10)

4.5 Käyttötaparyhmän suunnittelu

Käyttötaparyhmää suunniteltaessa on otettava huomioon laiteryhmän toiminnallisuus ja laitekokonaisuuden järkevä jako ryhmiin. Järjestelmässä voi olla useita käyttötaparyhmiä.

Ohjausjärjestelmäselostukseen (eng. Control System Description, CSD) kootaan suunnitteluvaiheessa tiedot laitteiden nimistä, moduuleista ja käyttötaparyhmistä tehtaan pohjapiirustuksen mukaan. CSD:n avulla käyttötaparyhmän kuljetinmoduuleille saadaan nimet, sekä moduulityypit.

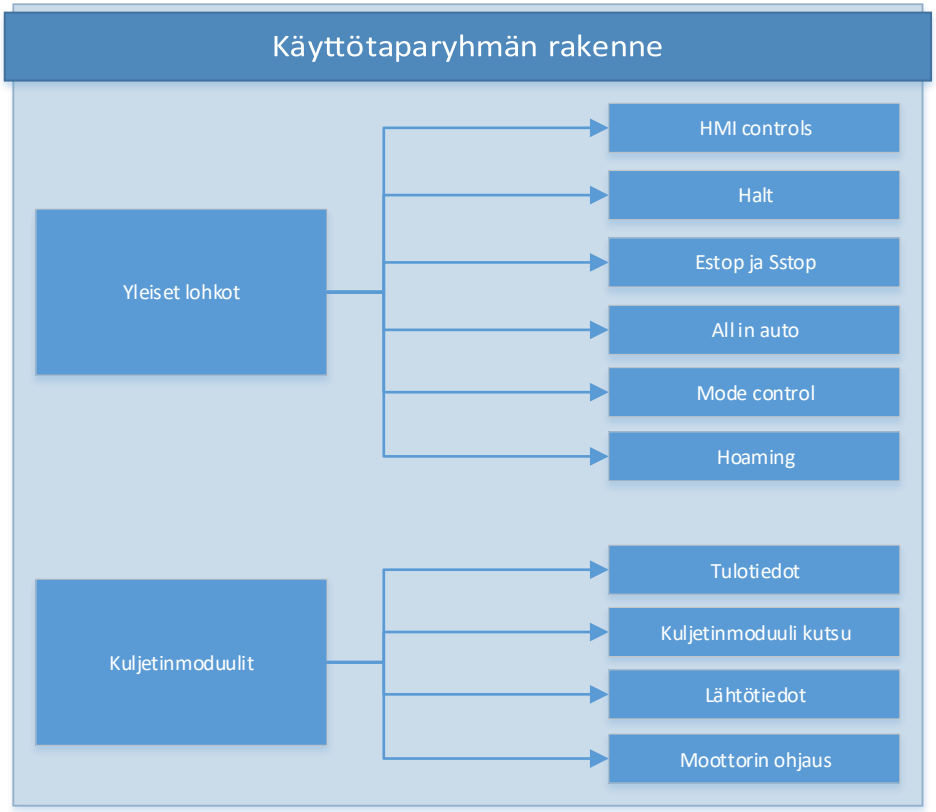
Ohjausjärjestelmäselosteen pohjalta täytetään Excel-tiedosto, johon kerätään kaikki käyttötaparyhmien ohjelmaa koskevat tiedot.

4.6 Käyttötaparyhmän kokonaisrakenne ja automatisointi

Käyttötaparyhmän kokonaisrakenne ja sen automaattinen luonti vaativat riittävän alustusselvityksen, jossa selvitetään käyttötaparyhmän sisältämät positiot suunniteltavasta järjestelmästä, niiden käyttämät kuljetinmoduulit sekä moduulien tarvittavat tiedot.

Käyttötaparyhmä koostuu eri osa-alueista ja jokaisen käyttötaparyhmän alussa käsitellään yleisten lohkojen tietoja, jotka koostuvat käyttötaparyhmän ohjaavista elimistä. Kaikkia käyttötaparyhmän positioita ohjataan yhteisestä ohjauselimestä, jossa määritetään ajo-tilat käyttötaparyhmän laitteissa. (Kuva 2.)

Kuva 2. Käyttötaparyhmän rakenne



Ohjauselinten jälkeen käyttötaparyhmässä tulevat laitteet. Rockwellin ohjauksen alaiset laitteet Cimcorpilla koostuvat suurimmaksi osaksi erilaisista kuljettimista, joko standardikuljettimista tai niiden erilaisista variaatioista. Mahdolliset muut toimilaitteet rakentuvat kuljetinmoduulien päälle tai toimivat omassa erillisessä PLC:ssä ja niiden toiminta ei ole suorassa vaikutuksessa käyttötaparyhmän luonnissa Rockwellin ympäristöön.

4.6.1 Yleiset lohkot

Käyttötaparyhmässä on joukko yleisiä lohkoja, jotka pysyvät rakenteeltaan vakiona riippumatta käyttötaparyhmästä. Näiden yleisien lohkojen eroavaisuus ryhmien välillä riippuu kuljetinmoduulien määrästä. Yleisiä lohkoja ovat toimitilojen ”SEIS”, ”KÄSIAJO” ja ”AUTOMAATTIAJO” –tilojen ohjaus, sekä toimitilojen ohjaukseen liittyvät inputit ja muut tiedot.

Työkalun luonnin kannalta yleiset lohkot tulevat olemaan yksinkertaisia kokonaisuuksia hallita, sillä ne ovat lähes vakiomuotoisia jokaisessa käyttötaparyhmässä.

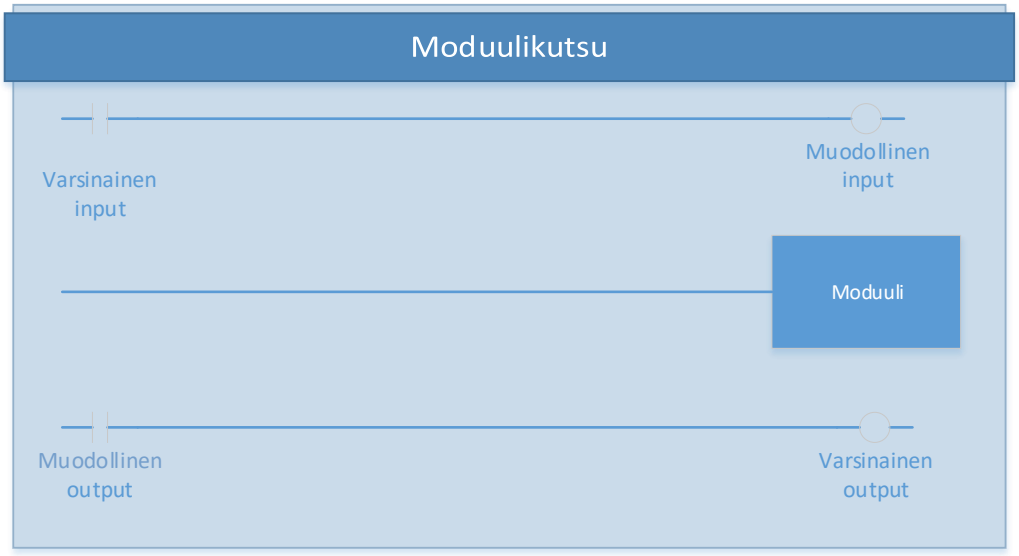
4.7 Kuljetinmoduulit

Kuljetinmoduulit ovat Cimcorpin standardoituja kuljetinohjauksia, joita käytetään valmiina kirjastoelementtinä tai pohjana erikoiskuljettimille, jotka ovat projektikohtaisia.

Käyttötaparyhmässä kuljetinmoduuleita käsitellään lohkoina, joita kutsutaan ohjelmakierron aikana. Kuljetinmoduulien lähtö- ja tulotiedot lisätään erillisinä piireinä (rung) käyttötaparyhmään.

Kuljetinmoduulin tulotiedot (input) ovat muodollisia inputteja ja käyttötaparyhmässä moduulin tuotavat inputit ovat varsinaisia inputteja, joilla varsinaiset tiedot tuodaan lohkoon. Lähtötiedoista (output) käytetään samaa järjestelmää (Kuva 3).

Kuva 3. Moduulikutsu



5 ROCKWELL STUDIO 5000

5.1 Studio 5000

Studio 5000 on Rockwell Automation / Allen-Bradleyn suunnittelema PLC-ohjelmistoympäristö, joka on suunniteltu yhdistämään suunnittelu ja ohjelmointielementit yhdeksi ohjelmistokokonaisuudeksi. Studio 5000 kokonaisuuteen sisältyy Studio 5000 Logix designer, Studio 5000 Architect, Studio 5000 View designer, Studio 5000 Logix emulate sekä Studio 5000 Application manager, johon kirjastointityökalut Library designer ja Library manager sisältyvät. Käyttötaparyhmän automatisointiin tarvitaan Logix designer ja ACM ja niitä käsitellään tässä työssä. (Rockwell automation, www-sivu)

5.2 Logix Designer

Logix designer on PLC-koodin luontityökalu, jossa varsinainen PLC-projekti luodaan, testataan ja ladataan logiikkaan. ACM ja Logix designer eivät linkity toisiinsa suoraan vaan Library designeriä ja Library manageria käytetään linkkinä ohjelmien välillä.
Logix designer ja Application code manager linkittäytyvät Library managerin välityksellä toisiinsa. Library manager on Logix designerissä lisäosana (Plug-in).

5.2.1 Ohjelmointikielet

Ladder logic eli tikapuukaavio on ohjelmakoodikieli, joka toteutetaan kytkentäkaaviona muistuttaen sähköpiirustusten kytkentäkaavioita. Ladder on suosittu PLC-ohjelmointikieli ja Cimcorp käyttää pääsääntöisesti sitä ohjelmointikielenä käyttötaparyhmissä, sekä muissa ohjelmalohkoissa.

6 ROCKWELL APPLICATION CODE MANAGER

6.1 Application code manager

Application code manager on PLC-koodien ja niistä luotujen kirjasto-objektien hallintaohjelmisto. ACM:n vahvuus on sen Excel liitännäisyys, jonka avulla saadaan massakopioitua haluttua kirjasto-objektia moneksi yksilölliseksi ohjelmaksi (program), rutiiniksi (routine) tai tehtäväksi (task). Kirjasto-objektien muokattavuus parametrien avulla on ACM:n ominaisuus, jolloin standardimuotoisten kirjasto-objektien käyttöönotto tapahtuu nopeammin, sillä tiettyjä parametreja, esim. nimiä voidaan muokata suoraan Excel-pohjasta, ilman varsinaiseen ohjelmakoodiin koskemista. (Rockwell, Application Code Manager User Manual 2019, 7)

6.2 ACM console

ACM console on ACM:n konsoliliitännäinen, jonka avulla voidaan ohjata ACM:ää samalla tavalla, mitä graafisen käyttöliittymän kautta voidaan normaalisti tehdä.

Konsolin tukemat komentokäskyt kirjoitetaan tekstitiedostoon.

Konsolin käyttämä XML-tiedostomuoto on kirjoitusasultaan normaalin XML-tiedoston muotoinen. Rockwellin käyttämät komennot ovat listattu dokumentissa: Logix 5000 controllers import/export.

ACM consolin käyttämä käskykanta on kapea ja sen päätarkoitus on tuoda valmiita kirjasto-objekteja projektiin, kirjasto-objektien massakopiointi Excelissä, kirjasto-objektien koodin generointi Logix designeriin sekä kirjasto-objektien HSL4-tiedostomuotoisien tiedostojen tuonti ACM:n ulkopuolelle. ACM:n ylläpitää tietokantaan kirjasto-objekteista, sekä niiden versiohallintaa. (Rockwell, Application Code Manager User Manual 2019, 85)

Taulukko 2. ACM konsolin komentokäskyt (Rockwell, Application Code Manager User Manual 2019, 85)

Komento	Kuvaus
Begincreate	Pakollinen aloituskomento ennen CREATE-komentoja
Createcontroller	Luo projektiin ohjaimen Extended Scritptin (.XML) pohjalta
Createobject	Luo ohjaimeen objekteja Extended Scritptin (.XML) pohjalta
Createproject	Luo uuden projektin Extended Scriptin (.XML) pohjalta
Delete; controller, project	Poistaa ohjaimen projektista tai poistaa projektin ACM:n tietokannasta
Editparameters	Muokkaa objekti instanssin parametreja
Endcreate	Pakollinen lopetuskomento CREATE-komentojen jälkeen
Exportallprojects	Tuo kaikki ACM projektit Excel-tiedostoiksi
Exportlibrariesby; Attribute, Project, Query	Tuo kirjastot HSL4-tiedostoksi muuttavan tiedon mukaan (attribuutti, projekti, kysely)
Exportpartial	Tuo osa ACM projektia Excel-tiedostoksi
Exportproject	Tuo projektin Excel-tiedostoksi
Generate; controller, partial	Generoi ohjaimen tai tietyn ohjelman/rutiinin L5X- tai ACD-tiedostoksi
Help	Näyttää kaikki komennot konsolissa
Importproject	Vie projektin ACM tietokantaan Excel-tiedostosta
Publishlibrary	Hakee ja julkaisee kirjaston ACD-tiedostosta
Registerlibrary	Julkaisee kirjaston ACM tietokantaan
Run	Aja konsoliohjelma
Switchdatabase	Vaihda ACM tietokannasta toiseen

7 TIEDOSTOTYYPIT STUDIO 5000

Studio 5000 sisältää eri tiedostotyyppejä ja tämän työn alueeseen kuuluvat tiedostotyypit ovat taulukossa 3.

Taulukko 3. Tiedostotyypit

Tiedostotyyppi	Ohjelmisto	Lisätietoa
.L5X	Logix designer	Kpl 6.1
.HSL4	Application code manager	Kpl 6.2

7.1 .L5X

L5X on Logix designerin käyttämä tiedostomuoto, joka on .XML-pohjainen kuvauskieli. L5X-tiedosto sisältää PLC-ohjelmakoodin XML-muodossa, jolloin sitä voidaan muokata millä tahansa tekstieditorilla ja koodin luonti esim. Excelin avulla on mahdollista. L5X:n erot normaalista XML-tiedostorakenteesta on siinä, että Rockwell käsittelee ([ ])-merkkien välistä dataa vapaaehtoisena. (Rockwell automation, 17456-RM084, 2018, 41) (Taulukko 4)

Taulukko 4. L5X-tiedostotyypin pääkomennot (Rockwell automation, 1756-RM084, 2018, 38)

Komponentti	Kuvaus	Koodi esimerkki
<RSLogix5000Content>	Kuvaa version ja vienti (export) tiedot	Liite 1
<Controller>	Kontrolleri/Ohjain	Liite 1
<DataTypes>	Käyttäjän määrittelemä ja I/O datarakenteet	Liite 1
<Modules>	Kontrollerin organisointi moduulit	Liite 1
<AddOnInstructionDefiniti-ons>	Lisäohjeet	Liite 1
<Tags>	Kontrollerin tagit	Liite 1
<Programs>	Ohjelmat	Liite 1
<Routines>	Ladder, FBD, SFC, ST-kielillä olevat rutiinit	Liite 1
<Tasks>	Kontrollerin tehtävät	
<ParameterConnection>	Parametrien yhteydet	
<ParameterConnections>	Ohjelman parametrien yhteydet	
<Trends>	Trendi, joka on konfigu-roitu projektin kontrolle-riin	
<QuickWatchList>	Kaikki QuickWatch-lis-tat, jotka on konfiguroitu projektiin	
<CommPorts>, <CST>, <WallClockTime>	Kontrollerin konfigu-rointi objektit	

Kaikki L5X import/export-tiedoston komponenttien rakenteet seuraavat samaa rakennetta:

<component_type [attribute_list]>
[body]
</component_type>
(Rockwell automation, 1756-RM084, 2018, 39) (Taulukko 5)

Taulukko 5. Komponentin rakenne (Rockwell automation, 1756-RM084, 2018, 39)

Esine	Selostus
<component_type>	Komponentin aloitustagi
Attribute_list	Lista atribuuteista, muodossa: attribute_name=”attribute_value”
Body	Komponentin sisältö, voidaan viitata eri alikomponentteihin tai muuhun tietoon komponentista. Valinnainen komponentti
</component_type>	Komponentin lopetustagi

Komponenttien selostukset rakentuvat standardista XML-tiedoston CDATA-elementistä:

”
<Task Name= ”Task1” Type=”PERIODIC” Rate=”100” Priority=”20” Watchdog=”500” DisableUpdateOutputs=”false” InhibitTask=”false”>
<Description>
<![CDATA[
This is a task description with a line feed and
“ a quote.
]]>
</Description>
</Task> “
(Rockwell automation, 1756-RM084, 2018, 42)

L5X:n ollessa XML-pohjainen, on tiedoston alussa oltava ylätunniste (Header):

<?xml version= <version_number> encoding= “UTF-8”
standalone= ”yes”/”no” ?>
(xmlwrite.net)

Taulukossa 6. Ylätunnisteen rakenne on selvennetty XML-ylätunnisteen rakenne.

Taulukko 6. Ylätunnisteen rakenne (xmlwriter.net, www-sivu, Rockwell automation, 1756-RM084, 2018, 39)

Esine	Selostus
<version_number>	Versionumero lainausmerkeissä, esim: “1.0”
Encoding	Koodinavaus, standardisoitu merkkiformaatti, esim. UTF-8.
Standalone	L5X-tiedostossa käytettään sisäistä DTD:tä, jolloin standalone-tietoon tulee ”yes”.

Ylätunniste tulee ennen varsinaisen ohjelmakoodia ja on XML-tiedoston peruskomponentti.

7.2 HSL4

HSL4-tiedostomuoto on Application code managerin käyttämä kieli, joka sisältää kirjasto-objektien XML-koodin. Tiedostomuoto muodostuu L5X-tiedostomuodon tavoin, lisäämällä <RSLogix5000Content>-komponentin tilalle kirjastointitiedot <Library> ja <CLX>. XML-sisältämä varsinainen ohjelmakoodi pysyy lähes muuttumattomana L5X:n ja HSL4-tiedostomuodon välillä. Riippuen Library designerissä tehtävistä muutoksista kirjasto-objektin tagien parametrointiin, muuttuu HSL4-tiedostoon parametrimuuttujien tiedot. HSL4-tiedosto sisältää myös tiedot kirjasto-objektin versiosta ja kirjastotiedoista, kuten mistä kirjastosta tiedosto on peräisin.

HSL4-tiedoston saa ACM:stä joko käyttämällä ACM:n graafista käyttöliittymää tai komentokäskyillä ACM:n konsolirajapinnan kautta.
HSL4-tiedostoa ei ole mahdollista tuoda ACM:stä ulos tekstieditoriin, muokata sitä ja viedä takaisin ACM:ään. Tiedosto on mahdollista tuoda ACM:stä ulos ja hyväksikäyttää sen sisältämää dataa L5X-tiedoston kokoamisessa. HSL4-tiedostorakenteessa ei ole Rockwellilta dokumenttia, tiedot on hankittu testaamalla ja tutkimalla Application code manageria. (Liite 2.)

8 KIRJASTOINTI

Kuljetinmoduulien siirtäminen nykyisestä kirjastosta Application code managerin sisällä olevaan Library manageriin ja sen hallinnoimaan kirjastoon tuo kirjastoinnin versiohallintaan automaattisuutta, sekä mukautuisi Cimcorpin kirjastointitapaan paremmin. ACM:n kuljetinmoduulien kirjastointi tulee olemaan välttämätön työkalun toiminnan kannalta. Library manager vaatii erillisen lisenssin, jotta sitä voidaan hyödyntää riittävän tehokkaasti. ACM lite-versio tukee vain paikallista tietokantaa ja kirjaston on oltava serverillä, jotta ohjelmoijat voisivat sitä tehokkaasti käyttää monelta päätteeltä.

Uusien ja nykyisten kuljetinmoduulien tallentaminen kirjastoon mahdollistaa työkalun ohjelmoinnissa riippumattomuuden kuljetinmoduulien sisällöstä kaiken muun osalta, paitsi lähtö- ja tulotietojen kanssa. Selvitettävä on miten ACM:n kirjastoitavista kuljetinmoduuleista parametrit saadaan asetettua, jotta työkalun luoma käyttötaparyhmä saa oikeat lähtö- ja tulotiedot linkitettyä. Toisaalta on tutkittava myös mahdollisuutta tuoda parametritiedot ACM:sta suoraan Exceliin, jossa niitä voitasi hyödyntää ohjelmallisesti helpommin.

9 OHJELMISTON KUVAUS

9.1 Tapa 1: Kirjasto-objekti

Alustavan teorian mukaan Logix designerillä ja Library designerillä luodaan alustava kirjasto-objektirunko, johon lisätään parametreja, joiden avulla kirjasto-objektista saadaan yleiskäyttöinen. Yleiskäyttöistä kirjasto-objektia käytettään ACM:n puolella pohjana yksilölliselle OMG:lle. Yleiskäyttöinen kirjasto-objekti ajetaan Exceliin, jolloin siitä muodostetaan .XML-taulukko, johon voidaan ajaa tietoa kuljettimista. Parametroinnilla myös kirjastoidut kuljetinmoduulit yhdistetään yksilöllisiin OMG:hin ja lopputuloksena saatu ryhmä yksilöllisiä OMG:ta generoidaan ohjelmakoodiksi haluttuun projektiin Logix Designeriin. Tällöin työkalun toiminnallisuus ja tarkoituksenmukaisuus on saavutettu. Käytettävyyttä pyritään kehittämään mahdollisimman yksinkertaiseen suuntaan alusta asti ja se on yksi pääprioriteeteista. Mahdollisia esteitä ACM:n käyttämiselle ja OMG:n luomiselle kirjasto-objektin pohjalta on Library designerin ja ACM:n joustamattomuus kirjasto-objektin rakenteen kanssa.

9.2 Tapa 2: XML-generaattori

Toinen ratkaisu, jos kirjastoinnin kautta ei saavuteta riittävää toiminnallisuutta ja työkalun käyttäminen johtaa kohtuuttomaan työhön loppukäyttäjälle, jotta koodi saataisiin generoitua, on suoraan Exceliin luotava VBA-ohjelma, joka hakee tarvittavat tiedot (kuljetinnumerot, moduulinumerot, ym.) suoraan Excel-taulukosta ja tulostaa niistä .XML-pohjaisen L5X-tiedoston. Alustava suunnitelma on käyttää ACM:n konsolirajapintaa L5X-tiedoston viennissä Logix designeriin sekä kirjasto-objektien tietojen tuomisen Exceliin.

Excelin ja ACM:n välinen kommunikointi toteutetaan Batch-tiedoston avulla, jolla saadaan ajettua komentoja ACM konsolirajapintaan. ACM:ssä hoituva kirjastointi ja versionhallinta keskittyy kuljetinmoduulien hallintaan, koska kuljetinmoduulit ovat standardi-muotoisia objekteja.

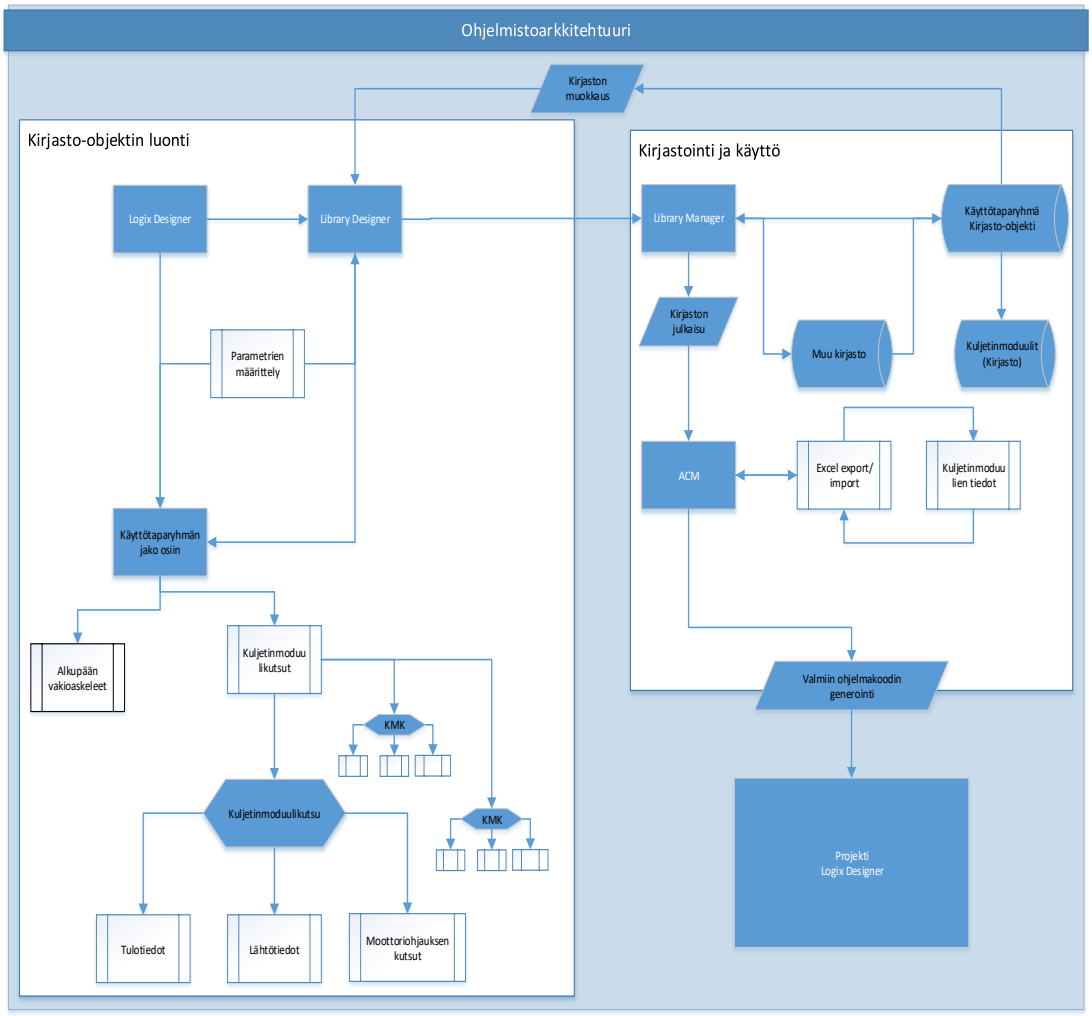
10 TAPOJEN TUTKIMUS

Lähdin selvittämään kahden eri toimintatavan käytettävyyttä ja toteutettavuutta ja sulkemaan huonomman vaihtoehdon pois ennen varsinaisen työkalun tekemisen aloittamista. Alustavasti lähdin selvittämään tavan 1 toteutettavuutta ja miten se vaikuttaisi nykyiseen ohjelmarakenteeseen.

10.1 Tapa 1

Tavan 1 kuvauksen mukaan käyttötaparyhmän luonti toteutettaisiin kirjasto-objektina mahdollistaen ACM:n käyttämisen ohjelman generointiin ja kirjastointiin sekä versiohallintaan.

Kuva 4. Ohjelmistoarkkitehtuuri



Application code managerin ja Logix designerin välinen kommunikointi toteutetaan kuvan 4 mukaisella tavalla, jossa käyttötaparyhmän runkorakenne luodaan Logix designerissä tuomalla se L5X-tiedostona olemassa olevasta projektista tai luomalla runkorakenne suoraan Logix designerissa.

Käyttötaparyhmän runko siirretään Library designeriin ja muuttujien parametrointi suunnitellaan ja toteutetaan. Kuljetinmoduuleista luodaan myös kirjasto-objekteja omaan kansioon. Parametrien avulla kirjasto-objektiin voidaan määritellä muuttujia, esim. moottorin nimi ja tunnus, jotka pysyvät standardimuotoisina jokaisessa samatyyppisessä kirjasto-objektissa. Parametrin arvo määritetään Excelissä tai ACM:n käyttöliittymässä.

Kirjasto-objektit tuodaan ACM:ään luotuun tietokantaan ja käyttötaparyhmäkirjasto-objekti viedään Exceliin, jossa siihen kirjoitetaan parametreihin laitetiedot. Tämän jälkeen Excel-tiedosto tuodaan takaisin ACM:ään ja tiedosto generoidaan koodiksi Logix designeriin.

10.1.1 Library Designer

Library designer on kirjasto-objektien luontityökalu, jonka avulla saadaan luotua ohjelmalohkoista yleiskäyttöisiä kirjasto-objekteja, jotka parametrien avulla kykenevät luomaan haluttuja kokonaisuuksia. Kirjasto-objektissa alustetaan kokoelma erilaisia ohjelmallisia funktioita, ehtoja ja tiloja, esim. tietyn kuljetinmoduulin ohjaus.

Kirjasto-objektin tageihin tuodaan ”koristeluja” (Decorations), eli tagit parametroidaan, jotta niihin saadaan ajettua haluttua dataa, esim. käyttötaparyhmän numero. Library designer suorittamat toiminnot:

- Luo kirjasto-objekteja.
- Määrittää kirjasto-objektin sisältämän Logix-sisällön.
- ”Koristelee” kirjasto-objektin parametreilla, aliobjekteilla, korvaavuuksilla, funktioilla, ulkopuolisilla vertauksilla, yhdistetyillä kirjastoilla ja rajapinnoilla.

- Luo korvaavuuksia merkkijonoihin, jotka vaikuttavat kaikkiin elementteihin kirjasto-objektissa instanssien luonnin yhteydessä.
- Luo matemaattisia ja loogisia ilmaisuja käyttäen koriste-elementtejä.
- Määrittää käyttäjän asettamat parametrit, funktioiden arvot tai viittaukset muihin elementteihin.
- Määrittää kirjasto-elementtiä koskevat sisällytykset instanssin luonnin yhteydessä.
- Luo tagit ja tagit-ryhmät ACM:n saataville parametrien avulla.
- Täyttää tagit parametrien, funktioiden ja ilmaisujen määrittelemänä.
- Julkaisee Logix-kirjasto-objektit suoraan ACM:n tietokantaan tai HSL4-tiedostoon.

Library designer kykenee luomaan standardi-muotoisesta ohjelmalohkosta kirjasto-objektin, jota voidaan hyödyntää muissa projekteissa. (Rockwell, Library Designer and Library Object Manager 2019, 10)

Kirjasto-objekti on lähtökohtaisesti standardi-muotoinen, jolloin sen sisältämän toiminnallisuuden tulisi olla riippumaton käyttökohteesta. Käyttötaparyhmän ohjelmoinnissa täyttä riippumattomuutta ei nykyisellä Cimcorpin käyttämällä ohjelmistorakenteella pystytä toteuttamaan, sillä Library designer käsittelee kirjasto-objekteja kiinteinä kappaleina/laitteena eikä ohjelmana. Kiinteänä laitteena kirjasto-objektin käsittely tarkoittaa, ettei sen perusrakennetta voida muuttaa. Cimcorpin käyttämä ohjelmistorakenne käyttötaparyhmässä muuttuu positioiden lukumäärän mukaan. Pääasialliset kuljetinmoduulikutsut ovat aina samanlaisia, erikoistapauksia lukuun ottamatta, mutta jokaisessa käyttötaparyhmässä on dynaamisesti muuttuvia ohjelmarivejä, joita ei voida pitää standardinomaisina. (Rockwell, Library Designer and Library Object Manager 2019, 10)

Library designerin alin taso kirjasto-objektin luonnissa on rung-taso, eli yhden askeleen taso. (Kuva 5). Jotta käyttötaparyhmästä voitaisiin luoda kirjasto-objekti, on alimman muokattavan tason oltava yksittäinen komponentti, esimerkiksi ladder-kielessä

kela ja mahdollisuus luoda tai muuttaa kelojen määrää tai sijaintia ohjelmassa. Nykyinen rung-tason muokkaaminen ja parametrointi ei mahdollista riittävää muokattavuutta kirjasto-objektiin.

Kuva 5. Rung

Kiinteän ohjelmanrunko-ongelman voi kiertää, esimerkiksi luomalla OR-funktion jokaiselle mahdolliselle muuttujalle oman kytkimen ja luomalla parametrin, joka arvossa lukuarvolla n tuodaan OR-funktioon n-määrä kytkimiä. Kyseisen tavan mukainen ohjelmistorakenne olisi yhteensopiva ACM:n tavalle käsitellä kirjasto-objekteja kiinteinä laitteina, mutta ohjelmallisesta näkökulmasta ratkaisu ei ole järkevä, sillä n-arvo voi kasvaa suuremmaksi, mitä sen maksimiarvo on kirjasto-objektissa. Ohjelmallisesti ei ole järkevää tehdä rakenteita, joilla on jokin fyysinen yläraja, jos sille ei ole perusteltua tarvetta, tässä tapauksessa perusteltua tarvetta ei ole, joten ohjerakennetta ei tehdä tavan 1 mukaan.

10.2 Tapa 2

Tavan 2 kuvauksen mukaan käyttötaparyhmän luonti toteutettaisiin Excelissä VBA-ohjelmointikieltä käyttäen mahdollistaen laajemman muokattavuuden käyttötaparyhmään. Cimcorpin käyttämä Excel-lähdetiedosto on merkittävä vaikuttaja VBA:n valintaan ohjelmointikieleksi tavassa 2. Cimcorp haluaa mahdollisuuksien mukaan pitää Excel-lähdetiedoston ja työkalun mahdollisimman yhtenäisenä pakettina, jolloin sitä on helppo liikutella yrityksen sisällä käyttäen hyväksi koko Cimcorpin kattavaa pilvitallennuspalvelua, joka hallitsee erilaisten dokumenttien versionhallintaa ja käyttöoikeuksia niiden muokkaukseen.

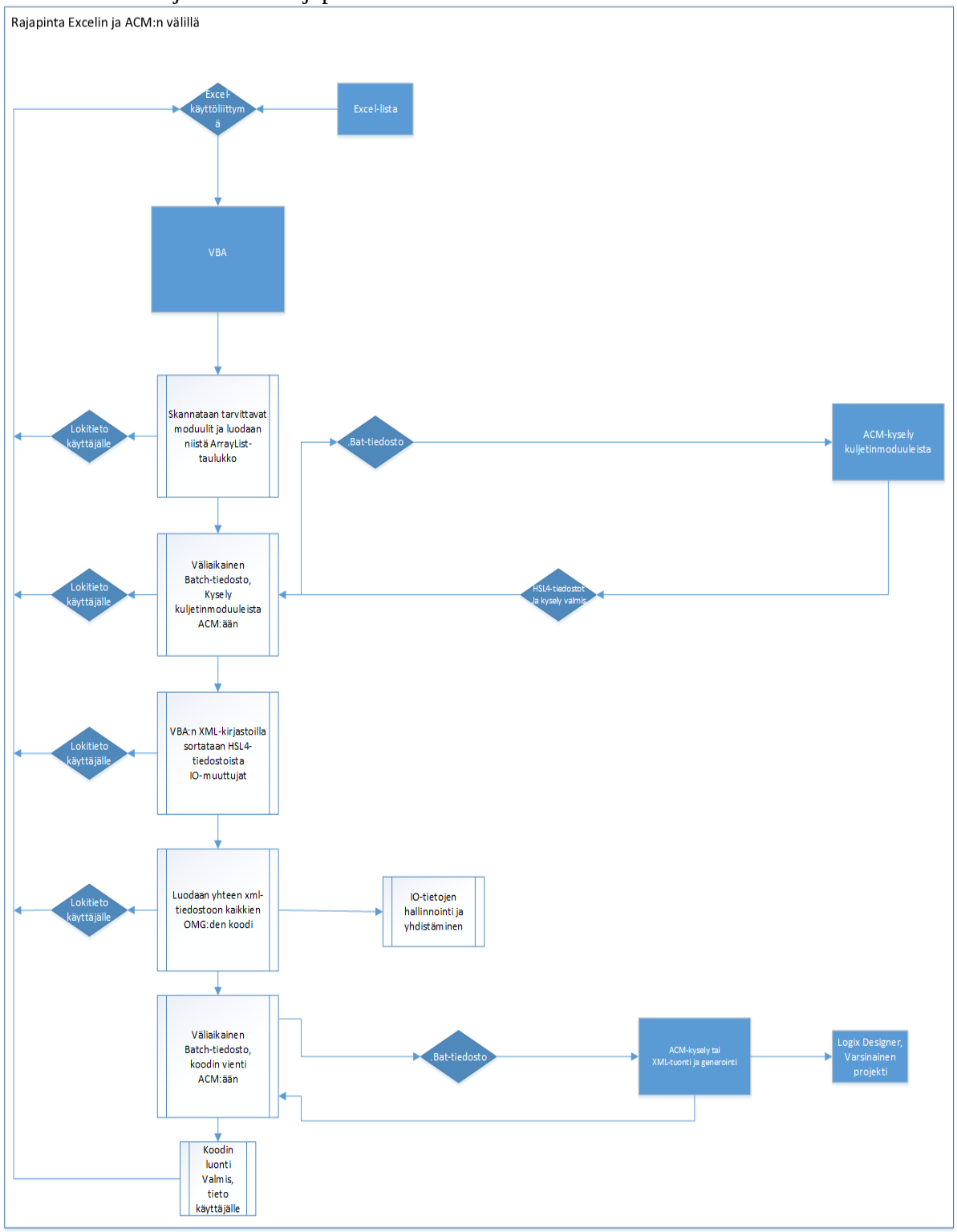
Logix designerin käyttämän L5X-tiedoston ohjelmarakenne jaotellaan osa-alueisiin VBA-ohjelmaa varten. Päärakenne, tagit ja datatyypit, rutiini ja rungit. Eri osa-alueista luodaan luokkia ja niitä käytetään olio-paradigmaa mukaillen. Tällöin vältytään ohjelmakierron pitkittymiseltä ja siirreltävän datan määrä vähenee. Samalla saadaan luotua

määräämätön määrä erilaisia käyttötaparyhmiä ja ohjelmaan ei tule rajoitusta luotavien käyttötaparyhmien ja niiden sisältöjen kanssa.

Cimcorpilla on olemassa oleva Excel-pohja, johon kerätään PLC-ohjelman suunnitteluvaiheessa tarvittavaa tietoa mm. kuljetinten käyttämistä moduuleista ja positionumeroista. Excel-pohja toimii työkalun tietokantana ja sen avulla käyttäjä pääsee vaikuttamaan helposti eri käyttötaparyhmien raakaan dataan. Itse työkalun käyttöä varten samaan Excel-pohjaan tehdään erillinen graafinen käyttöliittymä, josta käyttötaparyhmän generointia on edullista seurata ja varsinainen Excel-pohja pysyy tietokantana, eikä käyttöliittymänä itse työkalulle.

Application code manager toimii tavassa 2 tietokantana Cimcorpin standardikuljetinmoduuleille. Tällöin varsinaista Excel-pohjaa ei tarvitse päivittää aina standardikuljetinmoduulien muutosten yhteydessä, vaan työkalu käy noutamassa HSL4-tiedostot ACM:stä konsolirajapinnan välityksellä. HSL4-tiedostoista noudetaan tarvittava I/O-data, jonka avulla Excel-pohjan I/O-listasta ja kuljetinmoduulien kutsujen I/O-parit saadaan yhdistettyä L5X-tiedostoa varten. (Kuva 6.)

Kuva 6. Excelin ja ACM:n rajapinnasta.



Kuvassa 6 suunnitellun valmiin L5X-tiedoston vienti ACM:n konsolirajapintaa käyttäen Logix designeriin ei onnistu, sillä ACM ei tue ulkopuolisesta lähteestä tuotuja tiedostoja ja ACM:ään ei voi tuoda tiedostoa. ACM kykenee ainoastaan viemään

HSL4-tiedostoja pois ACM:stä ja käsittelemään jo kirjasto-objekteiksi luotuja tiedostoja ACM:n sisältämän tietokannan avulla.

Toinen ratkaisu ongelmaan oli yksinkertainen, mutta se vei osan työkalun automaatioasteesta pois. Valmiin L5X-tiedoston vienti Logix designeriin on työkalun käytön toinen manuaalinen vaihe Excel-pohjan täyttämisen ohella. Rockwell ei tarjoa Logix designeriin rajapintaa viedä ja tuoda ohjelmakoodia komentokäskyjen välityksellä, vaan L5X-tiedostojen vienti ja tuonti toteutetaan graafisen käyttöliittymän kautta Logix designerin sisällä.

10.3 Ohjelmistojen toimenkuvat XML-generoinnissa

10.3.1 Logix Designer

Logix designer toimii työkalun viimeisenä osana ja valmis L5X-tiedosto tuodaan käyttäen Logix designerin import L5X-työkalua, jolla saadaan tuotua ja generoitua tiedosto Ladder-ohjelmakoodiksi projektiin.

10.3.2 Library designer

Logix designerin ja Library designerin avulla nykyisistä standardikuljetinmoduuleista luodaan kirjasto-objekteja ACM:n tietokantaan. Moduuleja ei tarvitse muokata vaan ne voidaan nykyisessä muodossaan muuttaa kirjasto-objekteiksi. Kaikki muut datatyypit ja rakenteet, joista yksittäinen moduuli on riippuvainen, tulee kirjasto-objektin luonnin yhteydessä sisällyttää moduuliin tai tehdä kyseisistä datatyypeistä ja rakenteista omat kirjasto-objektinsa, jolloin versiohallintaominaisuudet koskevat niitäkin.

10.3.3 Application code manager

ACM toimii kirjastointityökaluna ja se ylläpitää standardikuljetinmoduulikirjasto-objekteja. Excelin ja ACM:n välinen liikennöinti toteutetaan Batch-tiedoston kautta. ACM palauttaa Excelille tiedostosijainnin, missä haluttujen kuljetinmoduulien HSL4-tiedostot sijaitsevat.

10.3.4 Excel

Excelissä oleva ohjelmointityökalun avulla toteutetaan käyttötaparyhmän koodin luonti käyttäen VBA-kieltä. Loppukäyttäjän täyttämän Excel-tiedoston pohjalta luodaan kysely ACM:ään standardimoduuleista ja generoidaan L5X-tiedosto ACM:stä ja Excelistä saaduista tiedoista.

10.3.5 Parametrien hallinta

Käyttötaparyhmän laitekutsujen muodollisiin input/output-tietojen yhdistäminen varsinaisiin pareihin suoritetaan alustavasti kahdella tapaa.

Lopullinen tapa on täysin automatisoitu, jossa työkalu hakee kirjasto-objektista muodolliset input/output-tiedot. Vertaamalla muodollisia input/output-tietoja Excelistä saatavaan I/O-listaan, ohjelma etsii oikean I/O:n jokaiselle parille. Pareihin, joihin ei tule suoraa I/O:ta antureilta, luodaan ohjelmistorakenteen mukainen pari. Tämä tapa vaatii I/O-listojen tekoon niin PLC-ohjelmoinnin, kuin sähkösuunnittelun nimeämiskäytäntöjen standardointia, jotta toiminto voidaan suorittaa automaattisesti. Mikäli nimeäminen vaihtelee projektista toiseen ja jopa projektin sisällä, ei nimivertailua voida tässä muodossa suorittaa, sillä pareja ei löytyisi.

Välivaiheen tapa ennen nimeämisstandardin kehittymistä on muodollisten ja varsinaisten input/output-tietojen nimeäminen Excel-listaan. Tällöin ohjelma hakee tiedon pareista Excel-tiedostosta ja sen perusteella toteuttaa saman parin luonnin mitä täysin automaattinen järjestelmä tekisi. Tässä tavassa loppukäyttäjältä vaaditaan Exceliin paritietojen esitäyttämistä, jokaiselle erilliselle positiolle tietoja ei tarvitse täyttää, vaan

tiedot täytetään kuljetinmoduulin tietosivulle. Tällöin kaikki positiot jotka käyttävät samaa moduulia, saavat samat positiokohtaiset tiedot pareihin.

Kirjastoitujen standardikuljettimien ja Excel-tiedostossa olevien varsinaisten input/output-tietojen kuvauksen versionhallinta hankaloituu, sillä yhden kirjasto-objektin muutoksen yhteydessä, myös Exceliin on muokattava kyseisen moduulin tietoja. Kahden tietokannan ylläpitäminen ei ole järkevää ja sitä tulisi välttää. Toisaalta on punnittava kuinka usein standardikuljettimien input/output-tiedot muuttuvat ja kuinka hyvin parien löytäminen saadaan toteutettua ohjelmallisesti siten, että varsinaiseen logiikkaohjelmaan ei tarvitse käydä tekemässä suuria muutoksia tai vaihtamassa suurta määrää input/output-tietopareja.

10.3.6 Käyttöliittymä

Työkalun käyttöliittymä rakentuu lähdetieto Excelin päälle ja sen avulla hallitaan työkalun toimintoja. Käyttöliittymän avulla työkalun käyttö tehdään suoraviivaiseksi ja Excel-tiedoston ja työkalun käyttö saadaan selkeästi eroteltua toisistaan.

Käyttöliittymän suunnittelussa on otettava huomioon eri osa-alueita:

- käyttöliittymän selkeys
- luettavuus
- tunnistettavuus, yhtenäisyys ja erottuvuus
- kieli
- tarkoituksenmukaisuus
- tavoitesuuntautunut informaation jäsentely ja esitys

Toimintojen asettelu käyttöliittymän ikkunaan tulee olla loogisessa ja luonnollisessa järjestyksessä. Keskeinen informaatio vasempaan ylänurkkaan, toisiinsa liittyvät osat ryhmiin tai muuten lähelle toisiaan. Ylimääräisen informaation karsiminen ja jäsentely selkeyttävät käyttöliittymää. (Asmala 2018).

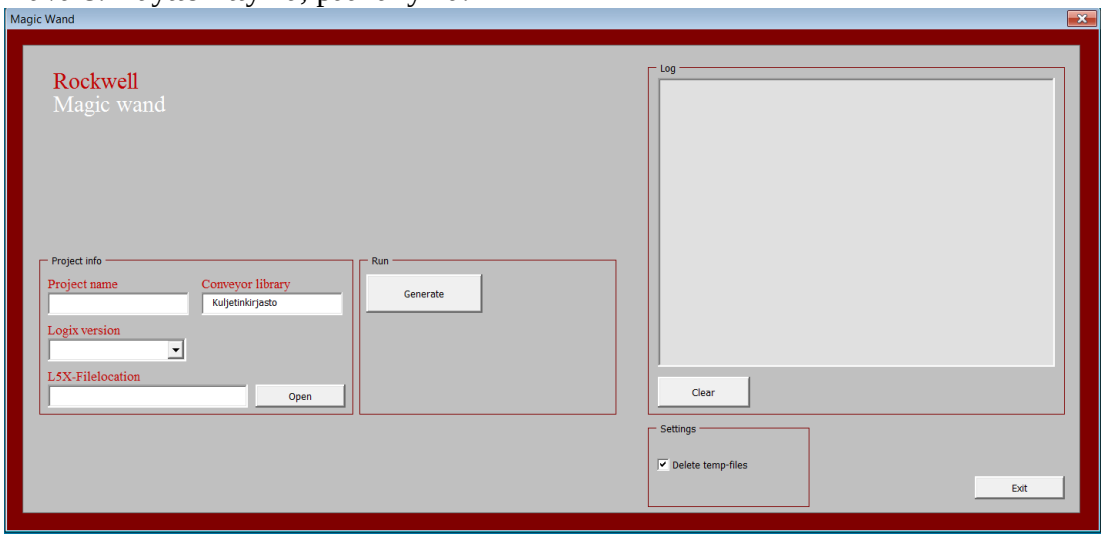
Käyttöliittymässä esitettävän tiedon koko, muoto, väri ja kontrasti vaikuttavat luettavuuteen. Räikeät ja liian suurella kontrastilla olevat osat ja elementit tekevät käyttöliittymästä epäselvän ja vaikeasti luettavan. Oikea värien valinta vaikuttaa myös käyttömukavuuteen, sillä hillityt värit ovat helpompia silmille, mitä suurella kontrastilla ja räikeillä väreillä olevat käyttöliittymät. Värien valinta on tärkeää, sillä käyttäjä voi käyttöliittymästä riippuen katsoa sitä pitkiä aikoja, jolloin on tärkeää ottaa huomioon käyttömukavuuteen vaikuttavia tekijöitä. (Asmala 2018)

Usein käyttöliittymissä on useita sivuja tai esitellään kirjo erilaista tietoa. Tiedon esilletuonti ja toiminnot tulevat olla tunnistettavia, yhtenäisiä ja niiden on erotuttava toisistaan selkeästi vähentäen virheen mahdollisuutta. Käyttöliittymän yleisiä periaatteita tulisi noudattaa näyttösivuja luodessa mahdollisimman paljon. (Asmala 2018).

Käyttöliittymän sanasto ja symboliikka tulisi olla sellaista, joka on tuttua käyttäjälle tai helposti yhdistettävissä käyttäjän jo omaavaan tietoon. (Asmala 2018).

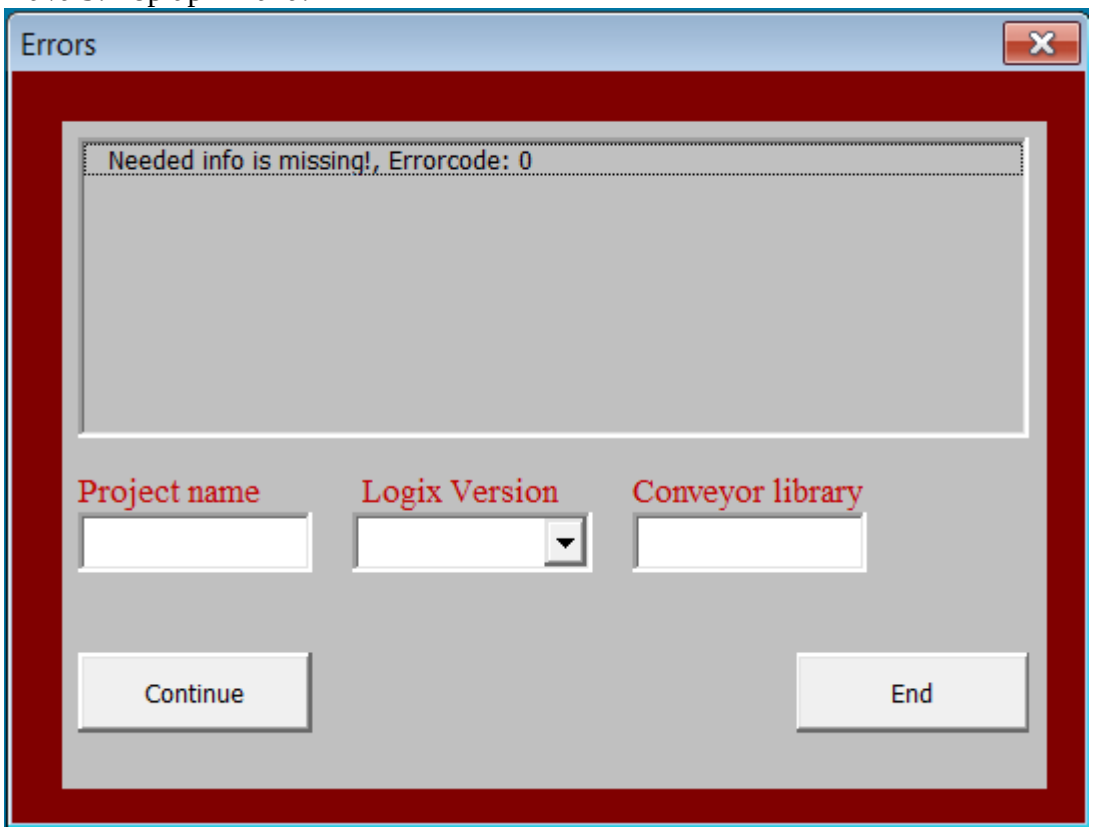
Käyttöliittymä, jolla työkalua hallitaan, muodostui yksinkertaiseksi käyttöliittymäikkunaksi (kuva 8), jossa käyttäjältä vaaditaan mahdollisimman vähän tiedon syöttämistä. Päänäkymässä on myös loki-kenttä, johon ohjelma tulostaa ohjelman eri vaiheiden suoritusten onnistumisesta. Käyttäjällä on myös mahdollisuus analysoida Batch-tiedoston sisältöä valitsemalla ”Delete tempfiles”-kohdan tyhjäksi. Tämä mahdollistaa mahdollisien virheiden analysointia ilman, että työkalun ohjelmakoodiin tarvitsee koskea.

Kuva 8. Käyttöliittymä, päänäkymä.



Pakollisten kenttien täyttö ilmennetään käyttäjälle pop up-ikkunana (kuva 9), silloin kun kentät jätetään täyttämättä ennen kuin työkalua ajetaan. Pop up-ikkunan tarkoitus on korostaa käyttäjälle, mitä tietoja tulee täyttää, jotta L5X-tiedoston generointi onnistuu. Ikkunassa kerrottaan käyttäjälle mikä virhe on kyseessä ja ohjeistetaan tietojen täytössä. Käyttäjällä on myös valinta lopettaa generointi, mikäli tietoja ei ole saatavilla. Mahdollisien virheiden ilmentyessä ne esitetään samalla pop up-ikkunalla, muuttaen näkymää sisältäen vain virhetiedot. Virhetiedon näyttäminen käyttäjälle vähentää ihmettelyä ja herättää käyttäjän huomion, mikäli virhe ilmenee työkalun ohjelman suorituksessa.

Kuva 9. Pop up-ikkuna.



Käyttöliittymän rakenteessa on otettu huomioon myös Cimcorpille opinnäytetyötä tekevän Aku-Petteri Partasen käyttöliittymän rakenne. Aku-Petterin opinnäytetyö käsittelee käyttötaparyhmien luontia Siemensin Tia Portal-alustalle. Molemmat opinnäytetyöt käyttävät samaa lähdetieto-Exceliä tietokantana työkaluille. Käyttöliittymien rakenne on pyritty pitämään mahdollisimman yhdenmukaisena, jolloin käyttäjän on helppo vaihtaa työkalusta toiseen, operointitavan pysyessä lähes samana.

11 TULOKSET

Tutkittiin kahden eri tavan mahdollisuuksia ja toimintatapoja käyttötaparyhmän PLC-koodin automaattista generointia varten. Cimcorpin käyttämä ohjelmistorakenne käyttötaparyhmissä asetti tiettyjä vaatimuksia työkalun toiminnalle. Varsinaista käyttötaparyhmän ohjelmistorakennetta ei lähdetty muuttamaan, sillä se olisi kasvattanut opinnäytetyön skaalaa liian suureksi. Tutkitun kahden eri työkalun toteutustavan välillä löytyi selvä voittaja.

Tavan 1 mukainen Application code managerin käyttö niin kirjastona kuin käyttötaparyhmien luontiin soveltuvana yleisenä työkaluna ei tavoittanut haluttua tulosta. Application code managerin tavasta käsitellä kirjasto-objekteja löytyi selvä heikkous puhuttaessa Cimcorpin käyttämästä ohjelmistorakenteesta. ACM ja Library managerilla ei ohjelmistojen nykyisellä tasolla (version 3.10.00, ACM ja Library manager tulevat samassa ohjelmistopaketissa) kyetty saavuttamaan haluttua toiminnallisuutta. Todettiin ACM:n käsittelevän kirjasto-objekteja komponentteina tai laitteina sen sijaan, että niitä käsiteltäisiin ohjelmallisina objekteina. Tämä käsittelytapa esti dynaamisten rakenteiden luonnin ilman suurten kovakoodaamisen ja fyysisten rajoitteiden asettamista ohjelmalle.

VBA:lla toteutettu työkalu todettiin mahdollistavan nykyisen Cimcorpin ohjelmistorakenteen käytön ja laajemman muokattavuuden itse työkalun ohjelmoinnissa, ottaen huomioon Cimcorpin käyttämän Excel-lähdetiedoston, johon kerätään merkittävä osa PLC-ohjelman ohjelmointia vaativista tiedoista. Vaikka kehittyneempiä ohjelmointikieliä olisi ollut paljolti saatavilla, päädyttiin VBA:han, sillä tällä tavoin Excel-lähdetiedosto ja varsinainen työkalu saatiin integroitua yhteen tiedostoon, jota on helppo liikutella ja muokata yrityksen sisällä.

Tapa 2 valittiin kehityssuunnaksi ja sen suunnitelman pohjalta alettiin ohjelmoida varsinaista työkalua Excel-lähdetiedostoon. IO-parien hallinta toteutettiin skannaamalla varsinaiset IO:t Excelistä ja muodollisien IO:den hakeminen standardimoduulien kirjasto-elementeistä. ACM:n rooli osoittautui odotettua pienemmäksi ja tavan 2 mukaan sen rooliksi jäi standardimoduulien kirjastointi ja versionhallinta.

Työkalulle suunniteltiin ja toteutettiin graafinen käyttöliittymä, jonka rakenteessa otettiin huomioon myös toisen samankaltaisen opinnäytetyön käyttöliittymä.

Opinnäytetyön tavoitteena oli tutkia eri vaihtoehtoja käyttötaparyhmän automaattiselle generoinnille ja rakentaa alustava ohjelamarunko käyttötaparyhmän generointityökalulle. Tuloksiin päästiin suunnitellun aikataulun sisällä ja ohjelmarungon ohjelmointi onnistui myös.

Satakunnan ammattikorkeakoulun automaatiotekniikkaan liittyvät kurssit auttoivat käyttötaparyhmän toiminnan ymmärtämisessä, sekä ohjelman suunnittelussa ja toteutuksessa. Eritoten kurssit: Automaatiotekniikka ja Käyttöliittymät olivat molemmat erityisen hyödyllisiä kursseja opinnäytetyötä tehdessä.

12 JATKOKEHITYS

Excel-lähdetiedosto sisältää kokoelman muita PLC-ohjelman luontiin liittyviä makroja, jotka tulevaisuudessa on mahdollista integroida osaksi käyttötaparyhmien työkalua ja hyväksikäyttää sen graafista käyttöliittymää. Makrojen integrointi käyttöliittymään tulee auttamaan makrojen toimintojen selkeyttämisessä ja käyttöohjeen laatimisessa. Graafinen käyttöliittymä jakaa varsinaisen Excel-taulukon ja käyttöliittymän selviin osiin, jolloin kokonaisuudesta tulee varsinaisen työkalun oloinen ja helpottaa uusien työntekijöiden oppimisprosessia.

Merkittävänä osana työkalun jatkokehitystä on integrointi Aku-Petteri Partasen opinnäytetyönä tekemään ohjelmaan käyttötaparyhmien ohjelman automatisoinnista Siemens-ympäristössä ja näiden kahden erillisen ohjelmiston sulauttamiseen keskenään. Molemmat on rakennettu saman Excel-lähdetiedoston pohjalle, mutta eivät ole samassa projektissa opinnäytetyön kirjoitusvaiheen aikana.

PLC-suunnittelun ja sähkösuunnittelun on luotava yhteinen nimeämisstandardi I/O-tiedoille, jotta käyttötaparyhmien automaattinen luonti voidaan tehdä. Ilman standardimallia ei PLC-koodin generointi ole käytännöllisellä tasolla mahdollista, sillä tällöin käyttäjälle jää suuri osa I/O-tiedoista manuaalisesti läpikäytäväksi. Tällöin PLC-koodin generoinnista automaattisesti ei saada hyötyä.

ACM:n tuomat ominaisuudet kirjastoinnissa ja L5X-tiedostoformaatti luovat tulevaisuuteen mahdollisuuksia automatisoida muitakin osa-alueita PLC-koodista. Esimerkiksi mahdollinen jatkokehitys työkalulle olisi valmiin projektipohjan luonti ja siihen liittyvien komponenttien generointi jo projektin alkuvaiheessa, jolloin projektissa olevat automaatioinsinöörit voisivat keskittyä projektikohtaisten erikoistapausten ratkaisujen tekemiseen. Samalla noussut automaation aste ja nopeampi PLC-koodin tuottaminen ylläpitävät ja valmistavat Cimcorpia jatkuvaan automaatioalan kehityksen mukana pysymiseen ja kilpailuedun ylläpitoon kasvavilla markkinoilla.

LÄHTEET

Asmala H 2018, Käyttöliittymät. Viitattu 25.7.2019.

www.moodle3.samk.fi

Cimcorp Oy 2019. www-sivut Viitattu 3.7.2019.

https://www.cimcorp.com/en/about-us/cimcorp-group

Rockwell Automation, www-sivu. Viitattu 25.6.2019

www.rockwellautomation.com

Rockwell, 17456-RM084, Logix 5000 Controllers Import/Export 2018, Viitattu 6.7.2019

www.rockwellautomation.com

Rockwell, Application Code Manager User Manual 2019, Viitattu 25.6.2019

www.rockwellautomation.com

Rockwell, Library Designer and Library Object Manager 2019, Viitattu 7.7.2019

www.rockwellautomation.com

SFS-EN ISO 13850, Koneturvallisuus. Hätäpysäytys. Suunnitteluperiaatteet 2015, Viitattu 3.7.2019. online.sfs.fi

Suomen Asiakastieto 2019. Viitattu 3.7.2019.

https://www.asiakastieto.fi/yritykset/fi/cimcorp-oy/18528284/taloustiedot

Suvela T 2010, Käyttötaparyhmä. Viitattu 1.7.2019

www.moodle3.samk.fi

xmlwrite.net. Viitattu 15.7.2019

https://xmlwriter.net/xml_guide/xml_declaration.shtml.

L5X-tiedostorakenne

HSL4-tiedostomuoto:

```
<?xml version=”1.0” encoding=”utf-8”?>
<Library [Attribute_data]>
        <Icon><![CDATA[]]><>/Icon>
        …
        <CLX [Attribute_data]>
                <DataTypes>
                        <Datatype [Attribute_data]>
                        …
                </DataTypes>
                <Programs>
                …
                </Programs>
                <Signature [Attribute_data]>
        </CLX>
</Library>
```