Tampereen ammattikorkeakoulu

Lasinkarkaisukoneen konfiguroinnin kehittäminen

Antti Ylä-Soini

OPINNÄYTETYÖ
Toukokuu 2023

Älyteollisuuden automaatioratkaisujen ylempi tutkinto-ohjelma

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Älyteollisuuden automaatioratkaisuiden ylempi tutkinto-ohjelma

YLÄ-SOINI, ANTTI:
Lasinkarkaisukoneen konfiguroinnin kehittäminen

Opinnäytetyö 61 sivua
Toukokuu 2023

Opinnäytetyön tarkoituksena oli selvittää Glaston Finland Oy:lle, kuinka heidän PLC-ohjelmansa konfigurointia voisi kehittää. Tarkoituksena oli selvittää ja antaa kehitysehdotuksia konfiguroinnin yksinkertaistamiseksi, nopeuttamiseksi ja laadun parantamiseksi. Tavoitteena on, että konfiguroinnin tulisi olla niin selkeää ja helppoa, ettei konfiguraatiosta huomaisi, onko sen tehnyt aloitteleva vai kokenut automaatiosuunnittelija.

Ongelman selvittämiseen käytettiin konstruktiivista tutkimusmenetelmää. Tutkimukseen haastateltiin Glastonilta työntekijöitä tuotekehityksen ja automaatiosuunnittelun puolelta, lisäksi pidettiin aiheesta useita palavereita. Tietoa kerättiin myös yrityksen sisäisestä epävirallisesta dokumentaatiosta tehtävää varten ja käytettiin työn tekijän vuosien kokemusta koneiden konfiguroinnista ja suunnittelusta. Aiheesta etsittiin tietoa myös kirjallisuudesta ja internetistä vahvistamaan teoreettista puolta.

Tutkimukset sisälsivät paljon nykyisen rakenteiden ja parametrien ryhmittelyä ja jäsentelyä. Haastattelut toivat paljon näkökulmia esiin, joista oli hyvä lähteä hakemaan ratkaisua esitettyihin ongelmiin. Kirjallisuudesta löytyi suoraan yllättävän vähän tosielämän esimerkkejä järjestelmien konfiguroinnista, mutta muutamalla esimerkillä saatiin avarrettua näkökulmaa, kuinka nykyisiin ongelmiin saataisiin käytännöllinen ratkaisu.

Lopputuloksena on monivaiheinen ohjeistus, kuinka konfiguraatiota voi vaiheittain lähteä kehittämään niin, että konfiguroinnin ylläpito, laatu ja helppous paranevat, eikä konfiguraation tekijän tarvitse olla enää kokenut konfiguroinnin asiantuntija.

Asiasanat: programming logic controller, konfigurointi

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Master's Degree Programme in Automation in Smart Industry

YLÄ-SOINI, ANTTI:
Developing Configuration of Glass Tempering Machine

Master’s thesis 61 pages
May 2023

The purpose of the thesis was to find out for Glaston Finland Oy how the configuration of their PLC program could be developed. The purpose was to find out and give development proposals to simplify, speed up and improve the quality of the configuration. The goal is that the configuration should be so clear and easy that one would not notice whether it was done by a beginner or an experienced automation designer.

A constructive research method was used to solve out the problem. Employees from Glaston were interviewed from the product development and automation design side, and several meetings were also held on the topic. Information was also collected from the company's internal informal documentation for the task and this thesis worker's years of experience in configuring and designing machines were used. Information on the subject was also sought from the literature and the internet to strengthen the theoretical side.
The studies included a lot of grouping and structuring of the current structures and parameters. The interviews brought out a lot of perspectives, from which it was good to start looking for solutions to the problems presented. In the literature, surprisingly few real-life examples of configuring systems were directly found, but with a few examples, a broadened perspective was obtained on how to get a practical solution to current problems.

As a result of the studies is multi-step instructions on how the configuration can be developed step by step so that the maintenance, quality, and ease of the configuration improves, and the person making the configuration no longer needs to be an experienced configuration expert.

Key words: programming logic controller, configuration

ALKUSANAT

Tämä työ on tehty Glaston Finland Oy:lle, jossa toimin automaatiosuunnittelijana. Työn tavoitteena oli keksiä, kuinka koneiden ohjelmallista konfigurointia voisi kehittää paremmaksi.

Haluan kiittää Glastonia joustavuudesta ja mielenkiintoisesta opinnäytetyön aiheesta. Glastonilta haluan kiittää erityisesti Sakari Palokangasta ja Jussi Eskolaa, jotka auttoivat tiedon keräämisessä ja haastatteluiden järjestämisessä.

Haluan myös kiittää oppilaitosta ja työn ohjannutta Jari Ruokolaista avusta ja tuesta opinnäytetyön kanssa.

Lisäksi haluan myös kiittää rakasta perhettäni tuesta ja kannustuksesta, jonka voimalla sain vietyä työn lopulta loppuun asti.

SISÄLLYS

LYHENTEET JA TERMIT

PLC	Programmable Logic Controller
XML	Extensible Markup Language
ADS	Automation Device Specification
UPS	Uninterruptible Power Supply
ERP	Enterprise Resource Planning
API	Application Programming Interface

1 JOHDANTO

Elämme informaatioyhteiskunnassa, jossa tietokoneet ja internet ovat suuressa roolissa. Tietoa löytää nopeasti ja sen pohjalta on helppo tehdä nopeita päätöksiä varsinkin, jos se tukee omaa näkemystä. Vielä muutama vuosikymmen sitten ihmiset olivat kärsivällisempiä kohdatessaan ongelmia, eikä heillä ollut mahdollisuutta vahvistaa omia huonoja kokemuksiaan tuotteesta. Tästä syystä internettiin kirjoitettu tieto jonkun tietyn asian ongelmista on erityisen haitallista yrityksille. (Leon 2015, 1.)

Tietokoneissa käytetyn ohjelmiston rooli on kasvanut viimeisen 10 vuoden aikana todella nopeasti. Ohjelmistosta on tullut jatkuvasti monimutkaisempaa, samalla ihmiset odottavat ohjelmiston käyttökokemukselta enemmän ja vertaavat niitä usein esimerkiksi Applen, Googlen tai Metan tuotteisiin, joissa toimintojen määrä, luotettavuus ja käyttökokemus on hiottu äärimmilleen. Samalla ihmiset ovat alkaneet vaatia enemmän. Ihmiset olettavat, että ohjelmat toimivat ilman ongelmia ja jos ongelmia tulee, he saavat siihen välittömästi apua ja että se korjataan hyvin nopeasti. Jos näin ei käy, ihmiset närkästyvät ja alkavat jakamaan negatiivista tietoa muille internetin käyttäjille kyseisestä palvelusta tai ohjelmasta. Vaikka vika korjattaisiin ja palvelu toimisi taas normaalisti, negatiivinen tieto asiasta jää internettiin ja sitä kautta aiheuttaa mainehaittaa yritykselle, vaikka ongelmaa ei enää edes ole. (Leon 2015, 1–2.)

Ohjelmiston monimutkaisuuden lisääntyessä testaus ja suunnittelun laatu nousevat suureen rooliin (Leon 2015, 2). Jos suunnittelu tehdään laadukkaasti ammattitaitoisten ihmisten toimesta ja testataan hyvin, virheellisen koodin mahdollisuus vähenee. Hyvin usein se ”paras” tekijä ei olekaan saatavilla suunnittelua varten ja testauskin jää vajaaksi, riski ohjelmiston ongelmille kasvaa. Lähes aina ohjelmiston suunnittelussa on kokemusperäistä ”hiljaista tietoa”, jossa jotain oleellista tietoa ei ole dokumentoitu mihinkään, vaan se on ainoastaan kokeneempien suunnittelijoiden yleisesti tiedossa oleva asia. Kävellessään ulos toimistosta viimeistä kertaa, nämä ammattilaiset vievät tiedon mennessään.

Tätä ongelmaa voi taklata myös tekemällä ohjelmasta määrätietoisesti selkeä ja helppo suunniteltava. Aina kaunis tavoite ohjelman pitämisestä selkeänä ja yksinkertaisena ei kuitenkaan toteudu ja ohjelman suunnittelu lakkaa noudattamasta alkuperäistä tavoitettaan. Tämä opinnäytetyö perehtyy yhteen esimerkkiin erittäin kompleksista koneenohjausohjelmasta ja sen käyttöön kehitetyn konfigurointityökalun uudelleen suunnittelusta ja kehittämisestä lähemmäs kohti alkuperäistä ideologiaansa.

2 LÄHTÖTILANNE

2.1 Yrityksen esittely

Tämä opinnäytetyö tehdään Glaston Finland Oy:lle, joka toimittaa lasin käsittelyyn ja prosessointiin tarkoitettuja koneita, työkaluja, ohjelmistoja ja toiminnanohjausjärjestelmiä sekä näihin liittyviä palveluita. Yritys toimii globaalisti ja sillä on tehtaita ja toimipisteitä useissa eri maissa. Yrityksen pääpaikka on Tampereen Vehmaisissa sijaitseva tehdas, jossa opinnäytetyön tekijä toimii opinnäytetyön kirjoitushetkellä automaatiosuunnittelijana.

2.2 Tutkimuksen taustat

Kehitystehtävä tehdään yrityksen tarpeesta kehittää ja standardisoida lasinkarkaisukoneiden automaatiosuunnittelun ja asennuksen läpivientiä. Näillä toimenpiteillä on tavoitteena saavuttaa tasalaatuisuutta samanlaisten, mutta eri koneiden välille niin, että koneesta voitaisiin löytää konekohtaiset prosessiin vaikuttavat asiat, eli ”koneen DNA”.

Glaston on jo useamman vuoden ajan kerännyt jokaisen asiakkaan erillisellä suostumuksella koneista käyttötietoa etäyhteydellä pilveen esimerkiksi tuotannossa käytetyistä ”resepteistä”, ajetuista laseista ja käyttöasteesta. Glastonin koneista keräämää dataa ja ”koneen DNA:ta” hyödyntämällä voitaisiin uusiin koneisiin myydä valmiita tuotannossa käytettäviä tuotantoparametreja, joilla asiakas voisi heti koneen luovutuksen jälkeen alkaa tuottaa huippulaatuista lasia.

Kehittämistehtävän aihe keskittyy automaatiosuunnittelun projektialustuksen koneen konfiguroinnin yleiseen kehittämiseen. Tavoitteena automaatiosuunnittelun projektien alustuksen kehittämisellä on saada nostettua projektien automaatiosuunnittelun laatua ja helpottaa eri koneiden konfigurointia niin, että virheiden mahdollisuus projektien alustuksessa minimoidaan ja alustus on tasalaatuisen laadukasta.

Kehittämistehtävän aihe on pala isompaa kehityskokonaisuutta, jota yrityksessä viedään nyt innokkaasti eteenpäin.

2.3 Kehittämistehtävän ongelma

Tavallisen karkaisukoneen ohjelmiston konfiguroinnissa on muokattavissa yli 5000 eri konekohtaista ohjelmaparametria, jotka vaikuttavat koneen käyttöön tai toimivuuteen, joten koneen konfiguroinnissa on syytä tietää, mitä parametreja pitää muuttaa, jotta kone toimisi oikein. Ongelma nykyisen konfigurointityökalun kanssa on se, että se tuo käyttäjän eteen lähes kaikki koneen parametrit. Parametrien suuren määrän takia parametrointi on todella sekavaa ja erityistä tarkkuutta vaativaa työtä. Lisäksi parametroinnissa on paljon sellaisia parametreja, joiden muuttaminen vaikuttaa moneen muuhun parametriin ja nämä ”säännöt” pitää vain tietää ja yrittää muistaa konfiguroinnin aikana. Tästä syystä koneiden konfiguroinnissa tulee usein huolimattomuusvirheitä.

Projektialustuksen laatuun vaikuttaa paljon suunnittelija ja suunnittelijan kokemus kyseisten koneiden konfiguroinnista. Projektien konfiguroinnissa vaaditaan kokemusta koneista ja eri koneiden toiminnallisista eroavaisuuksista. Projektien konfigurointia tekevät yrityksessä automaatiosuunnittelijoiden lisäksi myös Glastonin ”Automation Support” -tiimit Kiinassa ja Yhdysvalloissa. Tällä hetkellä haarakonttoreiden henkilökunnan suuren vaihtuvuuden ja kokemuksen puutteen takia haarakonttoreissa alustetut projektit joudutaan ainakin osittain käymään läpi myös Suomessa alustusten heikon laadun takia. Tämä syö turhaan resursseja Suomen konttorin valmiiksi kiireisestä suunnittelutiimistä.

Projektialustuksen aikana tehdyt virheet heijastuvat usein asennusaikataulun ja käyttöönoton venymiseen asiakkaan silmien alla ja pahimmassa tapauksessa aiheuttaa suuria ongelmia koneen tuottaman laadun kanssa pitkällä aikavälillä. Kaikki ongelmat koneen kanssa vaikuttavat negatiivisesti asiakaskokemukseen ja niillä on suuri merkitys silloin, kun asiakas päättää, ostaako hän uuden karkaisukoneen Glastonilta vai kilpailijalta.

Alla olevaan taulukkoon on kerätty aiheita, joihin toivotaan opinnäytetyöltä vastauksia:

TAULUKKO 1. Kysymykset, joihin halutaan vastauksia

Glaston tietää	Glaston haluaa tietää
Kun kaikki konfigurointiparametrit ovat esillä konfigurointityökalussa, konfigurointi on hidasta ja tarkkaa työtä, lisäksi virheitä tulee usein.	Kuinka parametrit pitäisi esittää, että ongelmat saadaan ratkaistua?
Toimilaitteiden lisääminen parametritiedostoon on haastavaa	Kuinka lisääminen tulisi toteuttaa, että se olisi helpompaa?
Parametritiedoston tuotantoon liittyvien parametrien arvot tulisi olla tuotehallinnan tiedossa ja ylläpidettävissä	Kuinka nämä parametrit voitaisiin esittää mahdollisimman helposti tuotehallinnalle ja tuotehallinta pystyisi tuomaan muutoksia alustustiedostoihin?
Parametrejä on todella paljon	Mitkä niistä on oleellisia?
Suunnittelu, alustus ja asennusvalvonta muokkaavat joskus samoja, mutta usein eri parametrejä	Voidaanko parametrejä luokitella käyttäjän mukaan?
Ohjelmassa on paljon toisistaan riippuvia parametrejä	Selvitys mitkä parametrit ovat toisistaan riippuvaisia ja millä tavalla? Voisiko joitain parametrejä korvata riippuvuuksilla?

2.4 Kehittämistehtävän tavoite

Tehtävän tavoitteena on luoda kehittämisehdotus, kuinka uuden parametrointityökalun tulisi toimia. Työn tarkoitus on saada nostettua projektien automaatiosuunnittelun laatua ja helpottaa eri koneiden konfigurointia niin, että virheiden

mahdollisuus projektien alustuksessa minimoidaan ja alustus on tasalaatuisen laadukasta.

Työssä tulisi lisäksi selvittää ainakin seuraavat asiat:

- mitkä ovat koneen konfiguroinnin kannalta kriittisiä parametreja
- Kuinka kriittisten parametrien muokkausnäkymä esitellään suunnittelijalle
- Tärkeiden parametrien sidossuhteet muihin parametreihin ja niiden muokkauksen automatisointi ohjelmalla
- Kuinka uusien parametrien tuodaan ohjelmaan
- Kuinka konfiguroinnista ohjelmalla saa tehtyä niin helppoa, että konfiguroinnin voisi tarvittaessa suorittaa joku muu kuin automaatiosuunnittelija

2.5 Tehtävän rajaus

Työssä esitellään nykyinen toimintaympäristö haasteineen ja tutkitaan, millä tavoin nykyistä ympäristöä voisi kehittää. Työssä otetaan huomioon vain työn tekemisen aikana käytössä olevat järjestelmät ja toimintamallit. Työssä annetaan kehitysehdotuksia ja ideoita, joiden pohjalta uudenlaista koneen konfigurointiin käytettävää ohjelmaa voisi lähteä suunnittelemaan.

3 KEHITTÄMISKOHTEEN KÄSITEYMPÄRISTÖ

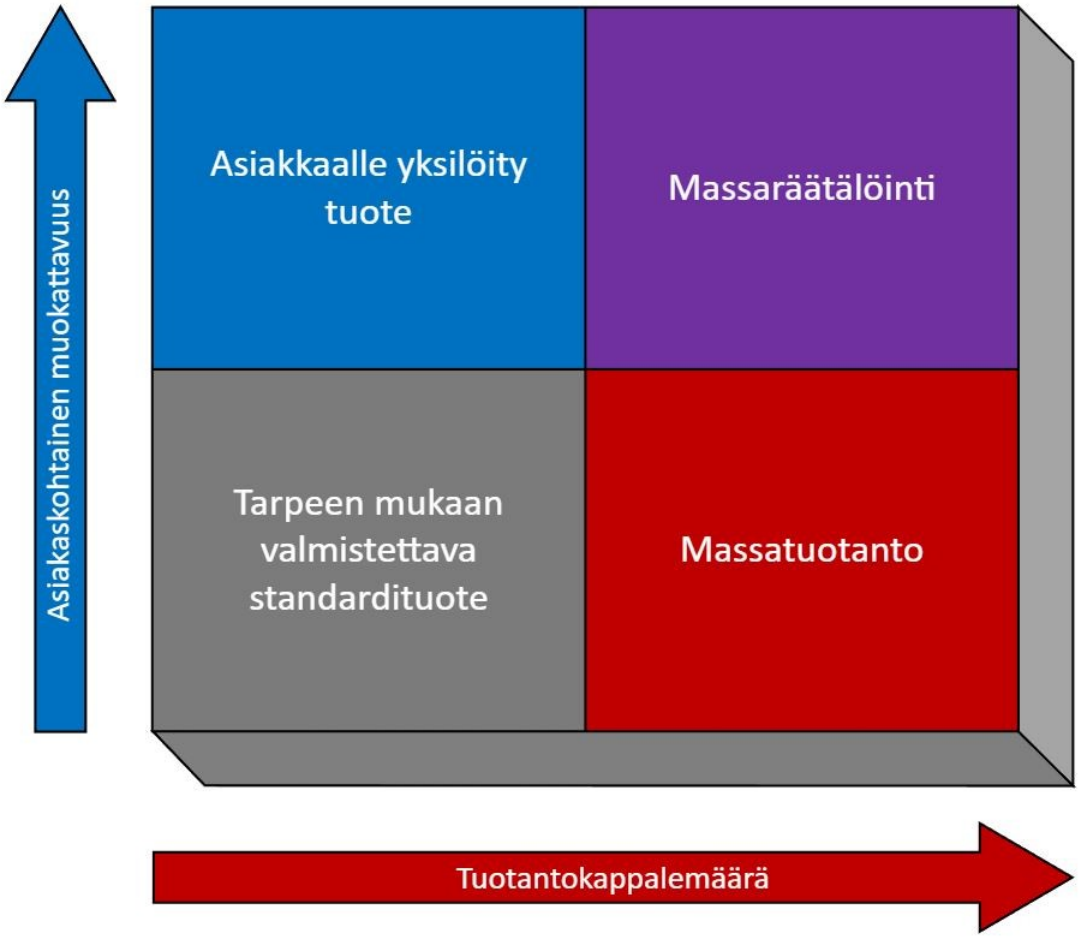
Seuraavaksi tutustutaan aiheeseen liittyvään kirjallisuuteen ja käydään läpi kehittämistehtävään olennaisesti liittyvä käsiteympäristö.

3.1 Tiedonkeräys kirjallisuudesta

3.1.1 Massaräätälöinti

Sana massaräätälöinti muodostuu kahdesta täysin vastakohtaisesta tuotantotavasta, massatuotannosta ja räätälöidystä tuotannosta.

Massatuotannossa tavoitteena on optimoida tuotanto tuottamaan paljon jotain tuotetta, jolloin yksittäisen tuotteen kustannukset laskevat mahdollisimman alas. Tämä mahdollistaa joko halvemman hinnan tuotteelle tai paremman katteen tuotteesta sitä myytäessä sekä yksittäistuotantoa suuremman kapasiteetin valmistaa tuotetta. Tämän tuotantotavan haittapuolena on vaikeus muokata yksittäistä tuotetta asiakkaan mieltymysten mukaan. Räätälöidyssä tuotannossa tuotetta tehdään asiakkaan mieltymyksien ja toiveiden mukaan, jolloin asiakas saa haluamansa tuotteen, mutta yksittäisen tuotteen kustannukset ovat massatuotantoa suuremmat sekä tuotteen läpimenoajat pidemmät. (Tseng, Wang & Jiao 2019)



KUVA 1. Massaräätälöinti

Massaräätälöinissä tuotekokonaisuus pyritään suunnittelemaan ja rakentamaan niin, että asiakkaiden toiveita tai vaatimuksia voidaan toteuttaa varsinaiseen tuotteeseen siten, että se ei vaikuta tuotteen läpimenoaikaan eikä lisää kustannuksia (Tseng, Jiao & Merchant 1996, 153). Massaräätälöinnillä pyritään tuottamaan asiakkaille räätälöityjä tuotteita massatuotantona ja saavuttamaan molempien tuotantotyylien parhaat puolet, räätälöidyn tuotteen muokattavuuden ja massatuotannon kustannus- ja kapasiteettihyödyt (Juuti 2008, 33). Näiden kahden hyödyn kokonaisvaikutus on merkittävä kilpailuetu markkinoilla, kun asiakkaalle voidaan tarjota yksilöllisesti heidän tarpeisiinsa räätälöityä ratkaisua edulliseen hintaan.

3.1.2 Tuotteiden konfigurointi

Tuotteiden konfigurointi on yksi tapa toteuttaa massaräätälöinnin ideologiaa. Tuotteiden konfigurointi koostuu tuotteiden ja tuoteperheiden muodostamisesta niin, että tuotetta asiakkaalle konfiguroitavaa tuotetta voidaan muutella asiakkaan mieltymysten mukaan tietyissä rajoissa siten, ettei muutokset vaikuta tuotteen läpimenoaikoihin tai kustannuksiin merkittävästi. Tuotteiden konfiguroinnin rajojen tulisi olla hyvin tiedossa, lisäksi erilaisista variaatioista tulisi olla jonkunlainen esisuunnitelma, jossa voitaisiin esimerkiksi hakea tuotteen skaalautuvuuden rajoja. Tuotteen konfiguroinnin suunnittelussa tulee olla hyvin perillä asiakkaiden tarpeista ja mieltymyksistä, mitä he haluavat heille myytävässä tuotteessaan muutettavan (Juuti 2008, 33–34). Konfiguroidun tuotteen läpimeno tuotannosta aina tilauksesta valmiin tuotteen toimitukseen ei tulisi vaatia erityistä suunnittelua, vaan kaikki tulisi tapahtua hyvin standardinomaisesti (Tiihonen & Soininen 1998).

Konfiguroitava tuote helpottaa myös myyntiorganisaatiota, koska konfiguroinnin rajoitusten ymmärtäminen selkeyttää, mitä myyjä voi asiakkaalle luvata ilman, että myyjän lupauksista koituisi huomattavia toimitus- tai kustannusongelmia (Jørgensen 2009). Myytäessä asiakkaalle uusia tuotteita, voidaan uusissa konetoimituksissa antaa kaupan päälle konfiguroitavia ohjelmallisia optioita, jotka eivät vaadi fyysisiä muutoksia koneeseen, jos se edesauttaa kauppojen syntymistä. Tällaisten konfiguroitavien ohjelmisto-optioiden käyttö on erinomainen keino viedä kauppoja maaliin, aiheuttamatta kumminkaan ylimääräisiä tuotantokustannuksia.

Isommissa kokonaisuuksissa ja pitkän elinkaaren tuotteissa, esimerkiksi tuotantolinjoissa modulaarinen ja pieniin palasiin pilkottu tuotteen rakenne helpottaa myöhemmin myös tuotteen muuttamista ja huoltotoimintoja. Esimerkiksi osa tuotteesta voidaan vaihtaa myöhemmin uudempaan paremmin asiakkaan käyttötarkoitukseen soveltuvaan vaihtoehtoon, jos asiakkaan toimintaympäristö muuttuu merkittävästi, esimerkiksi lainsäädännön takia.

Jos joku asiakas haluaa hänelle räätälöidyn ratkaisun, jota ei ole aiemmin tehty, yritys laskuttaa tuotteesta yleensä lisähintaa kattaakseen suunnittelukustannukset. Jos muutos nähdään tärkeäksi lisäksi tuotteeseen ja tehdään modulaarisesti,

voidaan ominaisuus tuoda konfiguroitavaksi tuotteeseen myös muille asiakkaille. Näin testattu ominaisuus voidaan myydä eteenpäin ilman että siitä koituisi merkittäviä suunnittelukustannuksia tai riskejä yritykselle.

3.1.3 Ohjelmistojen konfigurointi

Konfigurointi on tärkeä osa-alue modernia ohjelmistoa ja sen käyttöä. Ohjelmistokehityksessä ohjelma tehdään yleisesti tukemaan erilaisia ominaisuuksia, joita voidaan muokata käyttäjäkohtaisesti tai käyttötapakohtaisesti erillisen konfigurointitiedoston avulla. Konfigurointitiedostolla voidaan kertoa ohjelman käynnistyksen yhteydessä lähtötietoja varsinaiselle ohjelmalle ohjelman ominaisuuksiin, asetuksiin ja käyttäjäkokemukseen liittyen. Käyttäjä pystyy myös tallentamaan omia asetuksiaan ”muistiin” konfigurointitiedostoon, jolloin ohjelma muistaa seuraavan käynnistyksen yhteydessä esimerkiksi käyttäjän nimen. (What is a config file 2022)

Konfigurointitiedostosta on hyötyä, kun halutaan esimerkiksi siirtää konekohtaisia tietoja ohjelmasta toiseen, joko kaikille ilmaiseksi jaettavan tai palveluna myytävän ohjelmistopäivityksen yhteydessä. Käyttötarkoituksen mukaan, tiedosto voi olla sisällöltään hyvinkin yksinkertainen tai puolestaan sisältää todella paljon tietoa koneesta, kuten on esimerkiksi tämän opinnäytetyön tapauksessa, halutaan tehdä.

3.1.4 Ohjelmiston tuotteistaminen konfiguroinnilla

Kun puhutaan uuden ohjelmiston suunnittelusta, on siinä hyvin useasti tarkoituksena tehdä tuote, jota voidaan myydä asiakkaille. Näin kerran tehty ohjelma voidaan ”monistaa” muille, jolloin kerran tehdyn työn investointikustannuksia saadaan kompensoitua. (Hyvönen & Helokunnas 2003, 29–33)

Ohjelman suunnitteluun vaikuttaa asiakkaan personoinnin tarve ohjelmaa käyttäessä. Koska kyseessä on geneerinen ohjelma, eli samaa ohjelmaa käyttävät monet muutkin käyttäjät, personointia ei voi tehdä varsinaiseen ohjelmaan, vaan se

tulee tehdä erillisen tiedoston kautta, jota kutsutaan konfigurointitiedostoksi. Konfigurointitiedostoon voidaan tallentaa käyttäjäkohtaisia tai konekohtaisia asetuksia, jotka muokkaavat koneen käyttökokemusta asiakasta enemmän miellyttäväksi. (Hyvönen & Helokunnas 2003, 29–33)*)*

Ohjelmistotuotteen asiakaskunnan mukaan ohjelmistotuotteen myyminen asiakkaalle on hyvin erilaista. Kuluttajille myytävien tuotteiden tulee olla helppoja asentaa ja käsitellä, koska usein nämä ohjelmat ovat ilmaisia, eikä kuluihin ole budjetoitu asiakkaalle annettavaa palvelua tuotteen kanssa. Yksityisille kuluttajille myytävät tuotteen ovat usein edullisia tai ostettaessa ilmaisia, mutta sovellusta rahoitetaan näyttämällä sovelluksessa mainoksia, pyytämällä jonkun pienen kuukausimaksun tai käyttämällä sovelluksen sisäisiä mikromaksuja, joilla varsinaisesta ohjelmistotuotteesta saadaan asiakaskohtainen korvaus jollain aikavälillä. Näille tuotteille ominaista on myös asiakkaan pitäminen tyytyväisenä, esimerkiksi jatkuvilla päivityksillä ja uusilla ominaisuuksilla. Ei riitä, että tuote on hyvä, vaan sen pitää kyetä uusiutumaan ja tuottamaan asiakkaalle uusia elämyksiä. Kynnys, että kuluttaja vaihtaa palvelusta toiseen on matala, jos kuluttaja ei ole sijoittanut palveluun rahaa. Koska korvaus tällaisesta ohjelmasta yksittäiseltä käyttäjältä on usein myös hyvin maltillinen, näitä tuotteita tulee myydä paljon, että itse tuotteesta tai palvelusta saadaan kannattava.

Yrityksille ohjelmistoa myytäessä on hyvä huomata, että hyvin usein asiakkaan on tarkoitus tehdä ohjelmalla rahaa joko suoraan tai välillisesti. Tämä muuttaa asetelmaa siten, että asiakas on valmis käyttämään ohjelman ostamiseen rahaa tai maksamaan käytöstä lisenssi- tai ylläpitomaksuja. Tämä velvoittaa kuitenkin ohjelman suunnittelijaa joko tekemään ohjelmasta niin hyvän ja yksinkertaisen, että korjauksia ei tarvita tai sitten ylläpitämään ohjelmaa. Myös ohjelmistojen elinkaaret ovat (etenkin teollisuudessa) yleensä pidempiä ja päivittäminen hankalampaa. Yritysten välisessä kaupankäynnissä ohjelmistotuotteelle saatetaan joutua tekemään myös asiakaskohtaista räätälöintiä, jonka avulla asiakkaalle saadaan paremmin heidän käyttötarvettaan vastaava tuote. Asiakkaat ovat myös usein valmiita maksamaan tällaisesta palvelusta, jos se tuo heille tarpeeksi lisäarvoa.

Ongelma asiakaskohtaisissa ohjelmistoräätälöinneissä on yleensä ylläpidettävyys. Kun jollekin asiakkaalle tehdään oma versio jostain ohjelmasta, se täytyy

erottaa ohjelman normaalista ylläpidettävästä kehityksestä omaksi ”ohjelmistohaarakseen”. Tähän ohjelmistohaaraan tulee tuoda mahdolliset korjaukset ja uudet ominaisuudet käsin mahdollisesta kehityshaarasta, jolloin ohjelmistojen ylläpito muuttuu muutaman räätälöidyn projektin jälkeen raskaaksi. Ohjelmistojen käsin muokkaaminen ja päivittäminen vie aikaa ja jarruttaa yleistä kehitystä. Lisääntynyt työ sitoo entistä enemmän resursseja, jolloin joudutaan karsimaan muita töitä tai palkkaamaan lisäresursseja tai ostamaan kallista alihankintasuunnittelua. (Hyvönen & Helokunnas 2003, 25–26)

Tähän ongelmaan yksi ratkaisu on ohjelmiston konfigurointi. Jos ohjelmistosta tehdään konfiguroitava, asiakasräätälöinnit voidaan tehdä yhteen ohjelmahaaraan, jota parametroimalla räätälöinnit voidaan ottaa käyttöön tai käyttää vakioratkaisua. Jos joku haluaa samanlaisen räätälöinnin omaan ohjelmaansa myöhemmin, siitä voidaan pyytää lisämaksu, vaikka todellinen työ on tässä tapauksessa vain parametrin vaihto, koska ominaisuus on jo olemassa ja tehty kehityshaaraan. Jo tehtyjä räätälöintejä voidaan myydä vanhoille asiakkaille uusina ominaisuuksina, jolloin asiakkaille tuotetaan lisäarvoa olemassa olevien ratkaisujen avulla ja nostaa räätälöintien kannattavuutta. Tästä syystä ohjelmistoräätälöinnistä tulisi tehdä rakenteeltaan mahdollisimman geneerinen ja parametroitava, jotta se kävisi muillekin asiakkaille. Mitä useammalle valmis ominaisuus saadaan myytyä lisähintaan, sitä paremmaksi tuotteiden kate nousee.

Mitä laajempi tuotteen muokattavuus parametreillä on, sitä tarkemmin parametrointia kehittäessä pitää huolehtia, että vanhat parametrirakenteet tulee huomioitua uusia parametreja vanhoihin rakenteisin tehtäessä. Kun esimerkiksi konepajateollisuudessa myydään tuotantokone jollain ohjelmaversiolla ja se halutaan viiden vuoden päästä päivittää, ohjelman parametrirakenne on hyvin todennäköisesti muuttunut. Vanhan konfiguraation parametrien tulisi olla tuotavissa uuteen ohjelmaan niin, vanhoja muuttumattomia parametreja voitaisiin käyttää mahdollisimman kattavasti hyödyksi myös uudessa ohjelmassa. Tämä nopeuttaa ohjelman päivittämisprosessia ja poistaa turhia työnvaiheita. Lisäksi parametrien muuttumattomuus päivityksessä auttaa selvittelyissä, jos kone toimii jostain syystä, outosti tai virheellisesti päivityksen jälkeen. Ongelman aiheuttavaa ominaisuutta on helpompi hakea, mitä vähemmän muutoksia koneeseen on päivityksessä tullut.

3.1.5 RMS (Reconfigurable Machine System)

Eräs konfiguroinnin sovellus löytyy robotiikasta. Erilaisten RMS- tai RMT (Reconfigurable Machine Tool) - yhteensopivien työstökoneiden, kuten hitsausrobottien ja CNC-työstökoneiden muodostama kokonaisuus, jossa käyttäjä voi muokata konetta vapaasti riippuen käyttötarkoituksesta nopeasti ja kustannustehokkaasti konfiguroimalla kokoonpanoa. Käyttäjä pystyy konfiguroimalla määrittelemään koneelle uusia työkaluja ja lisäämään akseleita tarpeen mukaan. Perinteisellä työstökoneella eli DMS (Dedicated Manufacturing System) koneen muokkaaminen alkuperäisestä on erittäin rajallista ja FMS (Flexible Manufaturing System) järjestelmä on tuotannossa hitaampi ja kalliimpi RMS-järjestelmään verrattuna. RMS-järjestelmän huonona puolena on sillä tuotettujen tuotteiden tarkkuuden ylläpito. Panostamalla tuotteen laatua tarkkaileviin järjestelmiin tuotteen laatustandardia on kuitenkin mahdollista seurata ja ylläpitää. (Landers, Min & Koren 2001)

3.1.6 Yhteenveto kirjallisuudesta

Kirjallisuus tarjosi aika vähän konkreettista apua ongelmaan. Kirjallisuuden läpikäynti auttoi ymmärtämään kokonaisuutta paremmin ja toi uusia näkökulmia konfiguroinnin suunnitteluun, mutta loppujen lopuksi tarjosi hyvin vähän oikeita ideoita. Lähes kaikki käyttötapaukset, joissa konfiguroitavaa on paljon, on toteutettu seuraavanlaisella jaolla kolmeen ryhmään:

- Antamalla käyttäjälle konfiguroitavaksi kaikki parametrit
- Rajaamalla parametrit ”quick setup” tyyppiseen ryhmään, joka kokoaa yhteen useimmiten vaihdettavat parametrit
- ”Setup Wizard”, eli samanlainen step-by-step- työkalu, jota tässä ollaan nyt konfigurointiin soveltamassa

Yllättävän vähän tietoa löytyi itse konfiguroinnista. XML-Schema-standardi vaikuttaa olevan yleisesti käytetty konfiguroinnissa. Lähes kaikki tieto konfiguroinnista oli korkeamman tason ohjelmistokielien alustustiedostojen säädöistä, mutta mitään hienompaa logiikkaa näissä esimerkeissä ei vaikuttanut olevan.

Kirjallisuuden tutkiminen toi uusia näkökulmia työn toteuttamiseen. Massaräätälöinti ideana on hyvin pitkälti sitä, mitä koneen suunnittelussa ja ”tuoteperheiden” rakentamisessa on Glastonilla tähänkin asti käytetty.

3.2 Lasin karkaisu ja sen käyttö

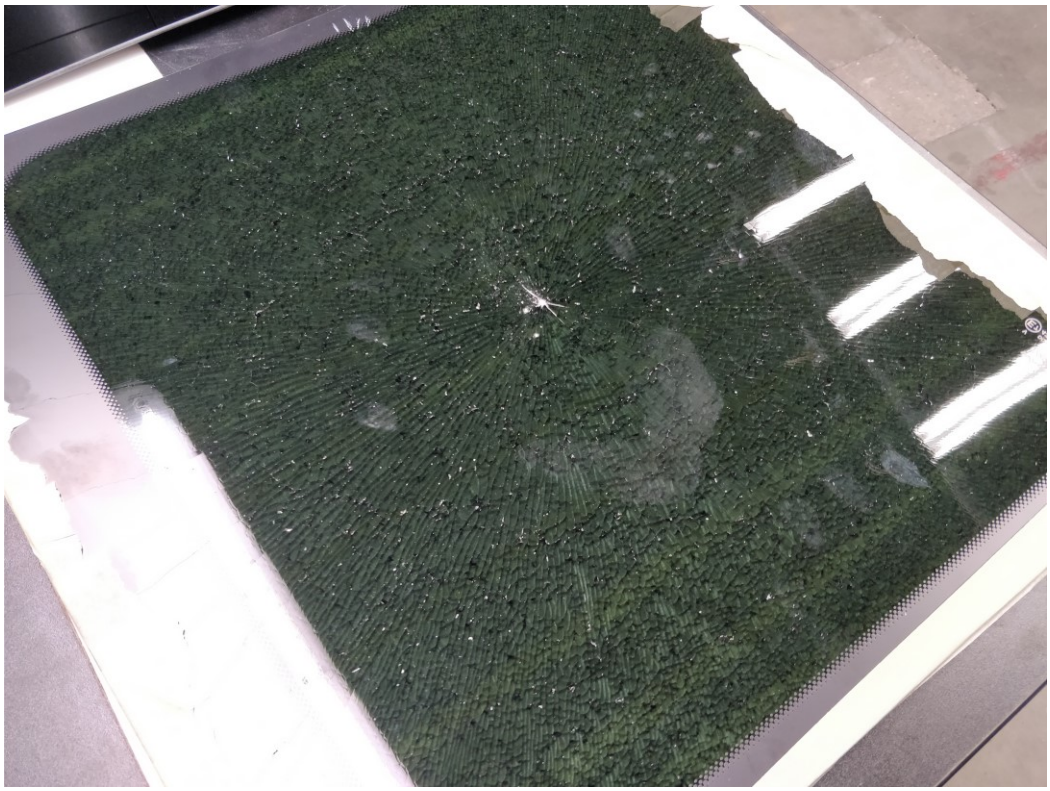
Järjestelmien kehittämisen kannalta olennaista on ymmärtää, millaiseen ympäristöön kehitystä tehdään. Esimerkiksi hektinen teollisuusympäristö tuo erilaisia vaatimuksia järjestelmien toimivuudelle, kun taas jos järjestelmää kehitettäisiin tavalliselle kuluttajalle. Seuraavaksi käydään läpi, millaiseen ympäristöön tämän opinnäytetyön kehitys tehdään.

3.2.1 Lasin karkaisun teoriaa

Lasin karkaisussa käytetään useimmiten Float-lasia, joka on yleisimmin käytössä oleva rakennuslasimateriaali. Lasia voidaan sellaisenaankin käyttää oikeaan kokoon leikattuna, mutta usein lainsäädäntö vaatii lasilta tiettyjä turvaominaisuuksia, joita tavallinen prosessoimaton lasi ei täytä.

Lasi voidaan karkaista joko lämpökäsittelyllä tai kemiallisesti. Lasin karkaisulla pyritään parantamaan lasin ominaisuuksia kestää paremmin iskuja, taivutusta ja lämmönvaihtelua. Lasin karkaisu nelinkertaistaa lasin veto- ja taivutuslujuuden. Kemiallinen karkaisu parantaa lasin vetolujuuksia, mutta hajoaa karkaisemattoman lasin tavoin, joten sitä ei voida luokitella turvalasiksi. Karkaisussa muodostuvien jännitteiden takia karkaistu lasi hajoaa eri tavalla kuin karkaisematon lasi. Normaalisti karkaisematon lasi saattaa halkeilla joko sivuista tai keskeltä menemättä kokonaan rikki. Kun lasi hajoaa palasiksi, palasen reunat ovat usein teräviä ja leikkaavia aiheuttaen ihmiselle vammoja. Karkaistun lasin jännitteiden takia

lasi on lujempaa eikä esimerkiksi halkeile reunoista. Jos lasin jännite rikotaan esimerkiksi terävällä iskulla lasiin tai taivuttamalla lasia liikaa, lasi hajoaa täysin pieniksi muruiksi. Murusien reunat ovat pyöreämpiä eivätkä pienemmän kokonsa takia aiheuta ihmisille yhtä vakavia vammoja kuin isot terävät reunat. Kaikki lasiin tehtävät mekaaniset toimenpiteet kuten leikkaus, reunojen hionta ja reiät tulee tehdä ennen karkaisua, koska karkaistu lasi ei hajoamisominaisuuksiensa takia kestä työstöä. (Rainamo & Riikonen, 20, 85–94)



KUVA 2. Karkaistu lasi hajoaa murusiksi

Lasin karkaisu on teoriassa helppoa, mutta suurimman ongelman karkaisussa aiheuttavat optiset muutokset lasissa karkaisun jälkeen. Tässä esimerkkejä karkaisuun liittyvistä laatuhaasteista:

- Vaakakarkaisussa telakannatuksesta syntyvä lasin ”aaltomaisuus”
- Oikeissa valaistusolosuhteissa tai polaroivien suodattimien kautta havaittavat kuviot lasissa tai laikukkuus
- Lasin fyysiset muodonmuutokset karkaisun aikana, kuten karkaistun lasin tai lasin reunojen taipuminen

Itse karkaisuprosessista johtumattomia ongelmia ovat myös lasin esikäsittelyn huono työnjälki ja karkaistavan lasin valmistuksen laatuongelmat, esimerkiksi lasin paksuuden liiallinen vaihtelu tai epäpuhtaat raaka-aineet. Vaikka lasi lämmitetään ennen karkaisua lähes sulamispisteeseen, lasin tulisi olla samanmuotoinen ja saman näköinen kuin ennen karkaisua. Poikkeuksen tähän tekee muotoon tai tiettyyn säteeseen tarkoituksella taivutetut lasit, joissa lasi muotoillaan uudelleen lämmityksen jälkeen ennen karkaisua, kun lasi on vielä pehmeää. Edellä mainittujen esimerkkien pohjalta lasin karkaisuprosessi luokitellaankin kokonaisuudessaan korkean teknologian osaamiseksi. (Rainamo & Riikonen, 20, 85–94)

3.2.2 Tasolasin karkaisuprosessi

Lasin karkaisu suoritetaan lämmittämällä lasi tasaisesti ensin noin 600 asteeseen ja sen jälkeen nopeasti jäähdyttämällä. Kun jäähdytyksessä kuuman lasiin puhalletaan suuria määriä ilmaa, lasi jäähtyy pinnoilta nopeammin kuin keskeltä, aiheuttaen pinnoille puristusjännityksen. Lasin jäähdyttyä huoneenlämpöiseksi nämä jännitteet säilyvät lasissa, joka antaa karkaistulle lasille sen ominaisen lujuuden. (Rainamo & Riikonen, 85)

3.2.3 Lasinkarkaisukone

KUVA 3. Karkaisukone (Glaston)

Lasinkarkaisukone on nimensä mukaisesti kone, jolla karkaistaan lasia. Kone on jaettu tavallisesti useisiin eri osiin työvaiheiden mukaan, joita kutsutaan myös ”osastoiksi” tai ”sektioiksi”. Näitä osastoja on peruskoneessa neljä: lastaus-, lämmitys-, jäähdytys- ja purkuosastot. Jokaisessa osastossa ja osastojen välissä lasia liikuttaa vaakasuorat koneen pituussuuntaan nähden poikittain olevat telat. Osastot ovat kiinni toisissaan niin, että lasi voi kulkea teloja pitkin kuljettimelta kuljettimelle ongelmitta. Koneen toiminta voidaan jakaa karkeasti neljään eri vaiheeseen:

1. Lastausosastossa lasi asetetaan lastauspöydälle joko käsin, avustimilla tai erillisillä lisäkuljettimilla. Kun lastaus on valmis, se kuitataan ja lähetetään teloja pitkin uuniin.

2. Uunissa lasi liikkuu keraamisilla teloilla edes takaisin, jota kutsutaan ”oskilloimiseksi”. Lasin edestakaisella oskilloinnilla mahdollistetaan lasin mahdollisimman tasainen lämpiäminen noin 600-asteeseen. Lasi lämmitetään lähelle sulamispistettä, joten lasin edestakaisella liikuttelulla estetään lasin muodonmuutokset. Lämmityksen jälkeen lastaus siirretään teloja pitkin jäähdytysosastoon.

3. Jäähdytysosastossa suuret karkaisupuhaltimet puhaltavat ilmakanavien kautta ilmaa kovalla paineella kuumalle lasille, jolloin lasi jäähtyy nopeasti. Kun lasi on jäähtynyt noin 50 asteiseksi, lasi kuljetetaan teloja pitkin purkuosastoon.

4. Purkuosastosta lasi siirretään käsin tai nostimilla pois purkupöydältä tai lasi jatkaa matkaa kolmannen osapuolen kuljettimelle ja kohti jatkokäsittelyä.

3.2.4 Karkaistun lasin käyttö

Karkaistua lasia käytetään moniin erilaisiin sovelluksiin, kuten rakennusten lasitukseen, ikkunoihin, kaiteisiin ja oviin. Vuonna 2018 muuttuneiden säädösten mu-

kaan kaikki alle 700 mm korkeudella lattiasta olevat lasit tulee olla turvalasia. Lisäksi jos lasin hajoamisen seurauksena on putoamisvaara, lasi tulee olla turvalasia. (Suomen Tasolasiyhdistys ry 2021)

3.2.5 Turvalasi

Turvalasiksi luokitellaan karkaistu, laminoitu tai karkaistu ja laminoitu lasi. Laminoinnissa lasien väliin asennetaan elastinen muovikalvo, joka sitoo kaksi lasia yhteen niin, että lasi ei hajotessaan anna periksi, vaan pysyy kehyksissään. Kun lasi karkaistaan ja laminoidaan, yhdistyy karkaistun lasin kestävyys ja laminoinnin sirpaleita sitova ominaisuus ja murronkestävyys. (Suomen Tasolasiyhdistys ry 2021)

Laminoinnissa voidaan käyttää myös puolikarkaistuja eli lämpölujitettuja laseja. Lämpölujitetun lasin ominaisuuksiin kuuluu karkaisematonta lasia parempi kyky kestää iskuja, taivutusta ja lämmönvaihtelua. Hajotessaan lasi kumminkin hajoaa isoiksi palasiksi, mutta palasten reunat ovat puolikarkaisussa syntyneiden jännitteiden takia tylpemmät kuin tavallisen lasin hajoamisen yhteydessä. Pelkkää lämpölujitettua lasia ei luokitella turvalasiksi. (Suomen Tasolasiyhdistys ry 2021; Tambest 2021)

3.3 Ohjausjärjestelmä

Seuraavaksi käydään läpi lyhyesti nykyiset järjestelmät ja ohjelmistoarkkitehtuuri.

3.3.1 Glaston PLC PC

Tämän opinnäytetyön kohdeyrityksessä Glastonilla on uusien lasinkarkaisulinjojen ohjauskäytössä PLC (Programmable Logic Controller) -PC. Glaston koneenohjauksessa käyttämä PLC-PC eli useimmiten Beckhoff- nimisen saksalaisyrityksen omilla komponenteilla suunnittelema teollisuus-PC, joka on suunniteltu kes-

tämää teollisuusympäristössä tavallista PC:tä pidempiä aikoja. PC itsessään sisältää tavallisen Beckhoffin räätälöimän Windows-ympäristön, joten PC:llä voidaan pyörittää tavallisia Windows-ohjelmia. Kun Beckhoff:in PLC ohjelmaa voidaan ajaa Windows-ympäristössä, ohjelma voidaan integroida todella vahvasti myös muihin Windows-ohjelmiin. Tämä mahdollistaa rajattomat mahdollisuudet ohjelmien käytössä PLC-ohjelman kanssa samassa tietokoneessa. (Scalable Industrial PC solutions 2021)

Beckhoff teollisuus-PLC:n käyttö Glastonin ohjausjärjestelmien suorittamiseen juontaa juurensa mahdollisuudesta suorittaa Microsoft Windows -käyttöjärjestelmää PC:ssä, sekä Beckhoffin oman CODESYS-pohjaisen, IEC 61131-3 PLC-standardiin pohjautuvan TwinCAT-ympäristön eduista. Vahvan Windows-sidoksen takia reaaliaikaohjelman käyttö ei ole rautasidonnaista, vaan ohjelmiston saa toimimaan missä tahansa Windows-tietokoneessa. Tämän lisäksi Windows-käyttöjärjestelmä mahdollistaa modernien tekniikoiden, kuten tietokantojen ja korkeamman tason ohjelmistojen käytön samalla PC:llä. Lähes kaikki sen kilpailijat, esimerkiksi Siemens ja Omron toimivat omilla PC:llään, joihin saa asennettua ainoastaan PLC-reaaliaikaohjelman, mutta ei muuta. Lisäksi näitä PLC-ohjelmia ei ole mahdollista suorittaa muussa kuin valmistajan määrittelemässä PLC:ssä, joten jos PLC-PC:n saatavuus heikkenee, ei valmistajan logiikkaa ole mahdollista korvata millään kolmannen osapuolen laitteella.

3.3.2 PLC-ohjelman konfigurointi Glastonilla

PLC PC:ssä pyörii Beckhoff TwinCAT- työkalulla tehty reaaliaikainen PLC-logiikkaohjelma nimeltään ”iControL”. Tämä iControL koneenohjaus-ohjelma on valtava sisältäen lähdekoodin koneen- ja prosessinohjaukseen lähes kaikentyyppisiin Glastonilla vuosien saatossa tehtyihin karkaisulinjakonevariaatioihin ja jopa joihinkin kilpailijoiden koneisiin. Monta erilaista konevariaatiota kattava ohjelmisto parametroidaan XML-kielellä kirjoitetulla konfigurointitiedostolla, joka sisältää koneen tyyppiparametreja, joilla voidaan kertoa PLC-ohjelmalle, millainen kone on kyseessä. PLC-ohjelma lukee XML-tiedoston ADS-rajapinnan kautta yrityksen omalta CONN-serveriltä, joka toimii linkkinä PLC-ohjelman ja ylemmän tason

käyttöliittymäsoftien kanssa. Asiakkaalle näkyvät käyttöliittymäohjelmat on toteutettu C#-kielellä, josta konetta voidaan käyttää ja hallinnoida. Tuotantoparametrit eli reseptit tallennetaan käyttöliittymän kautta Microsoft SQL-Serverille, josta niitä luetaan tarvittaessa. PC:ssä pyörii myös pilvi-client, joka lähettää koneesta dataa Glastonin pilveen datan jatkotarkastelua varten.

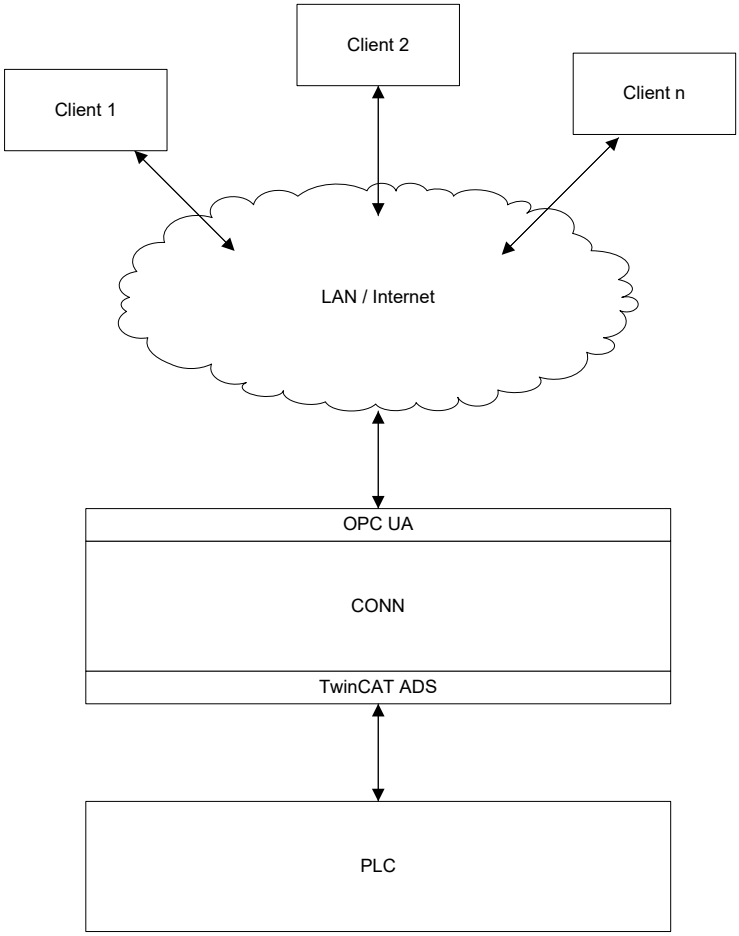
PLC-ohjelma tarkastaa annetut parametrit yksitellen läpi ja määrittelee ohjelman sisäisesti, minkälainen kone luodaan ja onko parametrointi looginen. Ohjelman sisällä on paljon konfiguroinnin tarkastuksia, joissa tarkastellaan konfiguraation loogisuutta ja onko se mahdollinen tehdä. Jos parametreista puuttuu ohjelman kannalta oleellisia parametreja tai ne on parametroitu ohjelmassa määriteltyjen arvojen vastaisesti, se pysäyttää parametrien läpikäynnin ja ilmoittaa siitä käyttäjälle CONN-serverin kautta viestillä, missä kohdassa parametrien tarkistus on pysähtynyt ja miksi. Virheviestin avulla käyttäjää opastetaan korjaamaan parametrointitiedostoa niin, että PLC toteaa konfiguraation loogiseksi ja hyväksyy sen.

Perus tuotantoprojektiin ei pääsääntöisesti tarvitse tehdä varsinaista PLC-ohjelmointia, vaan koneiden valmistelu asiakkaalle hoidetaan parametrointi-tiedostoa muokkaamalla. PLC-ohjelmaan kehitetään jatkuvasti taustalla versionhallinnan avulla uusia ominaisuuksia, joista tuodaan uusia parametreja uusiin PLC-ohjelmistoversioihin vähintään kerran vuodessa kootusti ohjelmistojulkaisujen kautta. Uusissa ohjelmistoversioissa uusia toimintoja tuodaan koneeseen uusien parametrien kautta ja niitä saa otettua käyttöön parametrointi tiedostoa muokkaamalla.

3.3.3 Glaston PLC PC:n kommunikointirakenne

PLC-logiikkaohjelman päälle kasattava kommunikointi koostuu kolmesta päätekijästä:

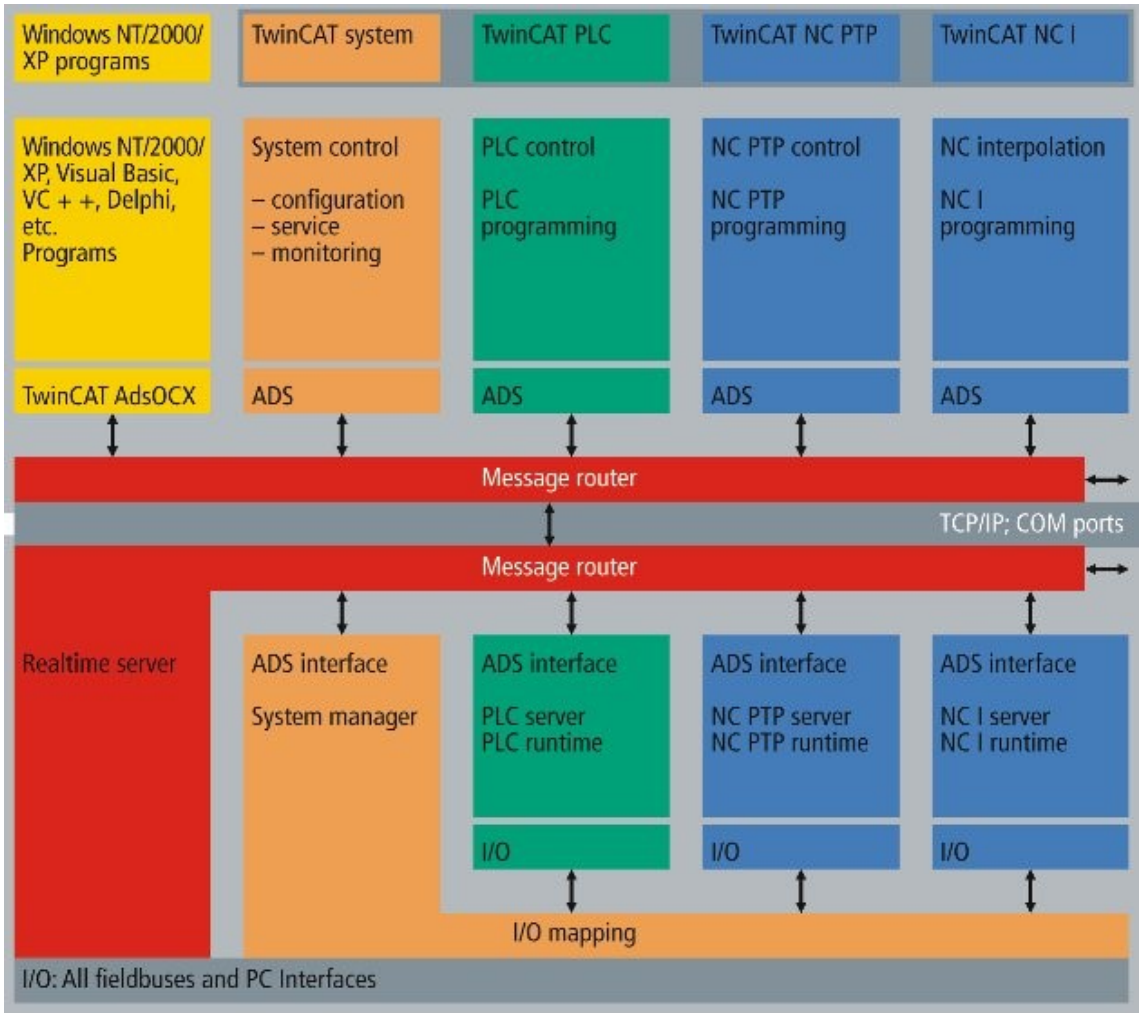
- TwinCAT ADS
- CONN-kommunikointiserveri
- OPC UA -rajapinta ulkoisesti liitettäviin järjestelmiin



KUVA 4. Tiedonsiirto Glastonin järjestelmässä

Seuraavassa käydään läpi välikerroksen järjestelmät yksi kerrallaan.

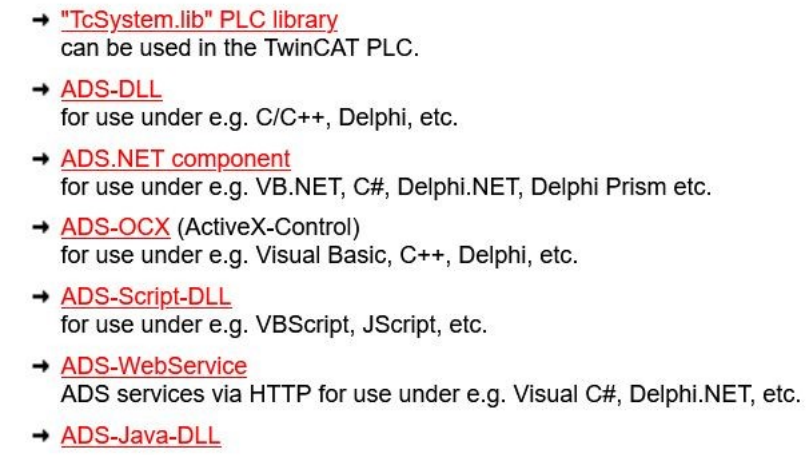
3.3.4 TwinCAT ADS



KUVA 5. Beckhoff TwinCAT:in kommunikaatio (TwinCAT ADS 2021)

Beckhoff:in omalla TwinCAT ADS (Automation Device Specification) –rajapinnalla on suuri rooli nykyisessä ympäristössä. TwinCAT-ohjelmassa ohjelman eri osa-alueet (PLC-ohjelma, IO-luenta, NC-ohjaus jne.) koostuvat ympäristössä omista ohjelmistaan, joiden välistä tiedonsiirtoa hoidetaan TwinCAT ADS-rajapinnan kautta. Tässä ”Message Routerissa” data siirtyy toimintojen välillä TCP/IP-yhteyksillä. (TwinCAT ADS 2021)

TwinCAT:in sisäiseen tiedonsiirtoon ADS-kommunikoinnissa päästään kiinni erillisellä ADS API:lla, joita löytyy eri ohjelmointikielille runsaasti:



KUVA 6. TwinCAT ADS API-kirjastoja (TwinCAT ADS 2021)

3.3.5 OPC UA

OPC UA (OPC Unified Architecture) on avoin alustariippumaton serveripohjainen kommunikointistandardi, jonka on kehittänyt OPC Foundation -järjestö, johon kuuluu yli 800 jäsenyritystä. Se on kehitetty yhdistämään eri tekniikoin tuotettua dataa ja mahdollistamaan mutkattoman tiedonsiirron standardiin kuuluvien sovellusten välillä. Rajapinnan kautta luettavassa datassa data on määritelty eri tasoihin ja selkeisiin rakenteisiin. Standardi on myös tietoturvallinen, koska datan luentaan ja kirjoitukseen voidaan määrittää erilaisia tasoja eri ohjelmille riippuen käyttötarkoituksesta. (OPC Foundation 2021)

Glastonilla OPC UA -rajapintaa käytetään tehtaissa yhdistämään PLC asiakkaan ERP (Enterprice Resource Planning) eli tuotannonohjausjärjestelmiin tai asiakkaan pilviserveriin. Rajapinnan avulla esimerkiksi PLC:n sisäiset muuttujat ovat luettavissa ulkopuolisten järjestelmien avulla niitä tarvitseviin sovelluksiin tai jonkun tietyn anturin dataa voidaan suoraan tallentaa pilviserverille.

OPC UA -rajapintaa käytetään myös kommunikointiin koneenohjaukseen käytettävän PLC PC:n ulkopuolisten sovellusten kanssa. Esimerkiksi korkeasti automatisoidussa tehtaassa OPC UA -rajapinnan kautta voidaan integroitua linjan muihin koneisiin, jolloin kommunikointi koneiden välillä hoituu OPC UA -rajapinnan kautta. Useimmiten tällaisissa linjoissa Glastonin linjaa edeltävä kone ”luovuttaa”

lastauksen Glastonin koneelle asiakkaan ERP-järjestelmästä saatavalla identifiointinumerolla ja reseptinumerolla. Glastonin kone tarkastaa reseptinumeron ja ottaa käyttöön reseptinumeroa vastaavan reseptin. Kun oikea resepti on valittuna, kone ottaa lasin vastaan lastauspöydälleen ja aloittaa prosessin automaattisesti. Kun lastaus on valmis, Glastonin linja lähettää lastauksen seuraavalle koneelle lastauskohtaisella identifiointinumerolla ja reseptinumerolla. Glastonin järjestelmä syöttää lastauksen valmistumisen jälkeen statistiikkaa lastauksesta asiakkaan ERP-järjestelmään OPC UA -rajapinnan kautta.

3.3.6 CONN-kommunikointiserveri

Glastonilla PC:n sisäisessä tiedonsiirrossa PLC-ohjelman ja ylemmän tason käyttöliittymäohjelmien kanssa käytetään CONN-kommunikointiserveriä. Se on periaatteessa PLC-koneella pyörivä OPC UA -serveri, joka toimii tiedonsiirtokerroksena yhdistäen PLC:n OPC UA:n ”AddressSpaceen”, jonka ”Client” ylemmän tasonkäyttöliittymä on. Tämän serverin kautta voidaan lukea tietoja ja tiloja PLC:ltä tai kirjoittaa PLC:n muuttujiin arvoja, jolloin käyttöliittymän ja PLC:n välille syntyy interaktiivinen käyttäjäkokemus. Tätä AddressSpacea käyttää muutkin koneen ylemmän tason käyttöliittymäohjelmistot. Tähän AddressSpaceen voidaan myöhemmin liittää Clientiksi OPC UA:n kautta asiakkaan järjestelmiä, jolloin näkymä näille Clienteille voidaan rajata haluamaksi OPC UA:n kautta.

3.3.7 XML-Schema

Glastonilla koneen konfigurointitiedostot kirjoitetaan XML-Schema-standardin mukaan. Tämä standardi on kansainvälisen W3C (World Wide Web Consortium) yhteisön määrittelemä, joka kehittää avoimia standardeja taatakseen internetin pitkä tähtäimen kehitystä (World Wide Web Consortium 2021). Tämä XML-Schema-standardi määrittelee selkeät säännöt sille, minkä rajoissa XML-tiedosto tiedosto voidaan tehdä niin, että ohjelma ymmärtää sitä. Rakenteessa määritetään, mitä XML-tiedosto saa sisältää ja mitä se ei saa sisältää. Tiedoston rakenteisiin tehtävän määrittelyn avualla voidaan tarkistaa, onko tieto määritelty tiedos-

toon oikein ja voiko sitä käyttää hyväksi. Jos tieto ei ole oikein, tiedoston lukeminen pysähtyy siihen suunnitellulla ohjelmalla virheeseen eikä tietoa voida lukea. Ohjelma antaa käyttäjälle virheilmoituksen korjata tiedosto kelvolliseksi, jotta se voidaan lukea. (XML tekninen ohjeistus 2021)

4 TUTKIMUSMETODOLOGIA

4.1 Konstruktiivinen tutkimusmenetelmä

Tutkimusmetodologialla tarkoitetaan lähestymistapaa tutkittavaan aiheeseen. Tässä opinnäytetyössä on ongelma luonteen takia valittu käytettävän konstruktiivista tutkimusmenetelmää. Konstruktiivisen tutkimusmenetelmän tavoitteena on löytää reaalimaailman ongelmiin looginen ratkaisu logiikan ja rakenteellisuuden kautta (Kasanen, Lukka & Siitonen 1993). Konstruktiivisessa tutkimusmenetelmässä korostuu empiirispainotteisuus ja teoreettisten johtopäätösten johtaminen ongelman ratkaisuun (Lukka 2000).

Ongelmana oleva käytettävän työkalun kehittäminen on selkeä reaalimaailman ongelma, johon kaivataan konkreettista ratkaisua. Tutkimus tehdään yhteistyössä kohdeyrityksen kanssa, jossa käydään aiheesta ideointipalavereja, jonka kautta pyritään ongelmaa ratkaisemaan. Tutkimuksen tuloksena on tarkoitus tuottaa kohdeyritykselle konstruktio, jonka tarkoitus on ratkaista alkuperäinen ongelma. (Lukka 2000)

4.2 Tutkimusmenetelmän käyttö opinnäytetyössä

Valitun tutkimusmenetelmän käyttö tässä kyseisessä opinnäytetyössä toteutetaan tunnistamalla ensin ongelma ja selvittämällä sen luonnetta ja siihen johtaneita syitä. Ongelman selvittämiseen haetaan ensin teoriapohjaista ratkaisua kirjallisuudesta ja internetistä, pääosin etsimällä konfigurointiin ja sen käsitteisiin liittyvää tietoa mahdollisimman laajalla näkökulmalla. Kerätystä kirjallisuudesta pyritään löytämään oleellinen tieto, joka auttaisi selvittämään tämän opinnäytetyön alkuselvityksissä ja haastatteluissa ilmenneitä ongelmia. Lopputuloksena pyritään löytämään looginen ehdotelma, jonka avulla havaitut ongelmat saataisiin ratkaistua.

4.3 Tiedon hankkiminen yrityksen sisältä

4.3.1 Palaverit

Alkuvaiheessa aiheesta pidettiin palavereja noin puoli vuotta parin viikon välein, jossa haettiin päälinjoja tutkimukselle ja täsmennettiin, mitä tutkimukselta ja konfiguroinnilta tulevaisuudessa halutaan. Palaverit kestivät yleensä tunnin ja paikalla oli tavallisesti automaatiosuunnittelun päällikkö ja käyttöliittymätiimin päällikkö. Lisäksi palavereissa vieraili niin tuotekehityksen työntekijöitä kuin muita automaatiosuunnittelijoita antamalla konsultaatiota aiheeseen liittyen.

4.3.2 Henkilökohtaiset tapaamiset

Henkilökohtaiset tapaamiset pidettiin usein automaatiosuunnittelun päällikön tai toisen automaatiosuunnittelijan kanssa joko kasvotusten tai Teams-yhteyden välityksellä. Tapaamisten aihe oli usein tarkempien teknisten yksityiskohtien läpikäynti aiheeseen liittyen. Henkilökohtaisissa palavereissa käytiin usein läpi myös PLC- ohjelmiston historiaa ja sitä, mikä sen suunnittelussa on hyvää ja mikä huonoa juuri konfiguroinnin kannalta.

4.3.3 Nykyisen järjestelmän dokumentaatio

Nykyisestä järjestelmästä on jonkin verran materiaalia saatavilla, mutta se pohjautuu pitkälti järjestelmän kokonaisuuden esittelyyn. Varsinaista dokumentaatiota konfiguroinnista ei ole, mutta ohjeita koneiden konfigurointiin löytyy.

4.3.4 Yhteenveto tiedon hankkimisesta yrityksen sisältä

Yrityksen sisäiset keskustelut aiheesta olivat antoisia. Haastatteluissa keskusteliin laajasti esimerkiksi työkalun historiasta, alkuvaiheen visioista, tulevaisuuden

suunnitelmista ja konkreettisista ongelmista työkalussa ja sen kehityksessä. Monessa kohtaa kävi ilmi, että Finanssikriisin (vuosina 2008–2009) aiheuttama taantuma ja liikevaihdon ja tilausten romahdus lopetti suurimman kehityksen kuin seinään, jonka jälkeen työkalun kehitys on takkuillut työvoima- ja rahallisten resurssien puutteen vuoksi. Lisäksi alussa määritellyt ohjelmalliset skaalautuvuudet eivät ole kyenneet pysymään erilaisten konetyyppien ja kokojen kasvun perässä.

Palaverien ja haastatteluiden tuloksia käydään tarkemmin läpi työkalun kehityksen yhteydessä myöhemmin opinnäytetyössä.

5 UUSI TYÖKALU

5.1 Konfigurointi Glastonilla

Glaston valmistaa karkaisulinjoja asiakkaillensa hyvin erilaisiin käyttökohteisiin: on asiakkaita, jotka ajavat pieniä laseja massatuotantona linjan läpi ja heille kapasiteetti on tärkein asia, joku asiakas taas ajaa isoja yksittäisiä laseja, ja jotkut ajavat jotain siltä väliltä. Jollekin asiakkaalle laatu on kaikki kaikessa, toiselle riittää, että yleiset turvallisuusstandardit täyttyvät, kunhan kapasiteetti on korkea. Yleisin tilanne on, että tuotannossa tasapainoillaan laadun ja korkean kapasiteetin välillä. Pitkään alalla olleilla yrityksillä on omaa visiota, kuinka he haluavat koneen toimivan ja tahtovat vahvasti heille räätälöidyn ratkaisun. Toisaalta mukana on myös asiakkaita, jotka ovat vasta aloittelemassa alalla eikä heillä esimerkiksi ole prosessiosaamista ja haluavat automatisoida kaiken mahdollisen. Usein miten ohjausjärjestelmä asennetaan uuteen linjaan, kun taas joku asiakas haluaa päivittää uusimman koneenohjauksen hyvin palvelleeseen, jopa yli 30 vuotta vanhaan linjaansa.

Vaikka Glaston pyrkii mahdollisimman pitkälle tuotteistamaan karkaisulinjansa niin suunnittelun kuin valmistuksenkin osalta, lähes aina myydyissä linjoissa on asiakaskohtaisia eroavaisuuksia. Nämä pienet muutokset ohjelmistossa koneiden välillä vaativat lähtökohtaisesti suunnittelua, testausta ja tarkkuutta, ettei koneessa esiinny vaaratilanteita tai asiakkaan tuotantoa haittaavia tilanteita.

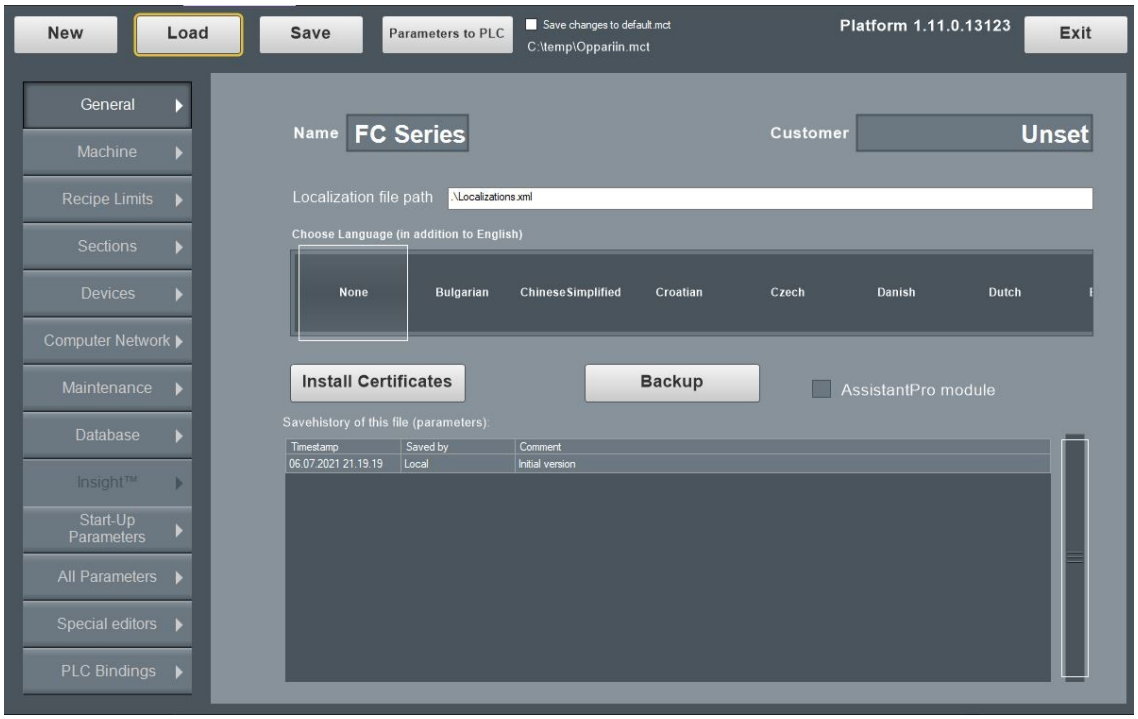
Nämä muutokset on toteutettu yleensä kahdella eri tyylillä, yksilöllisesti ohjelmakoodia räätälöimällä projektia varten tai sitten tuotu hallitusti ja suunnitelmallisesti mallisarjaan konfiguroitavaksi ominaisuudeksi.

5.1.1 Konfiguroinnin nykytila

Tavanomainen tuotantoprojektin automaatiosuunnittelu on pääosin koneen konfigurointia kahdella eri työkalulla: kenttäväylä I/O:n konfigurointia Beckhoff:in kehittämillä työkaluilla ja koneen XML-konfiguraatiotiedoston konfigurointia Glastonin omalla itsekehittämällä MCT-työkalulla (Machine Configuration Tool).

5.1.2 MCT – työkalu

MCT-työkalu on Glastonin itse tekemä koneenkonfigurointi XML-tiedostojen käsittelyyn tarkoitettu työkalu. Editori lukee koneen XML-tiedoston, käy sen rakenteen eheyden läpi ja sen perusteella joko avaa MCT:n editoriin muokattavaksi tai jos XML-rakenne ei ole ehyt, MCT-työkalu antaa käyttäjälle virheviestin.



KUVA 7. MCT-työkalun päänäkymä

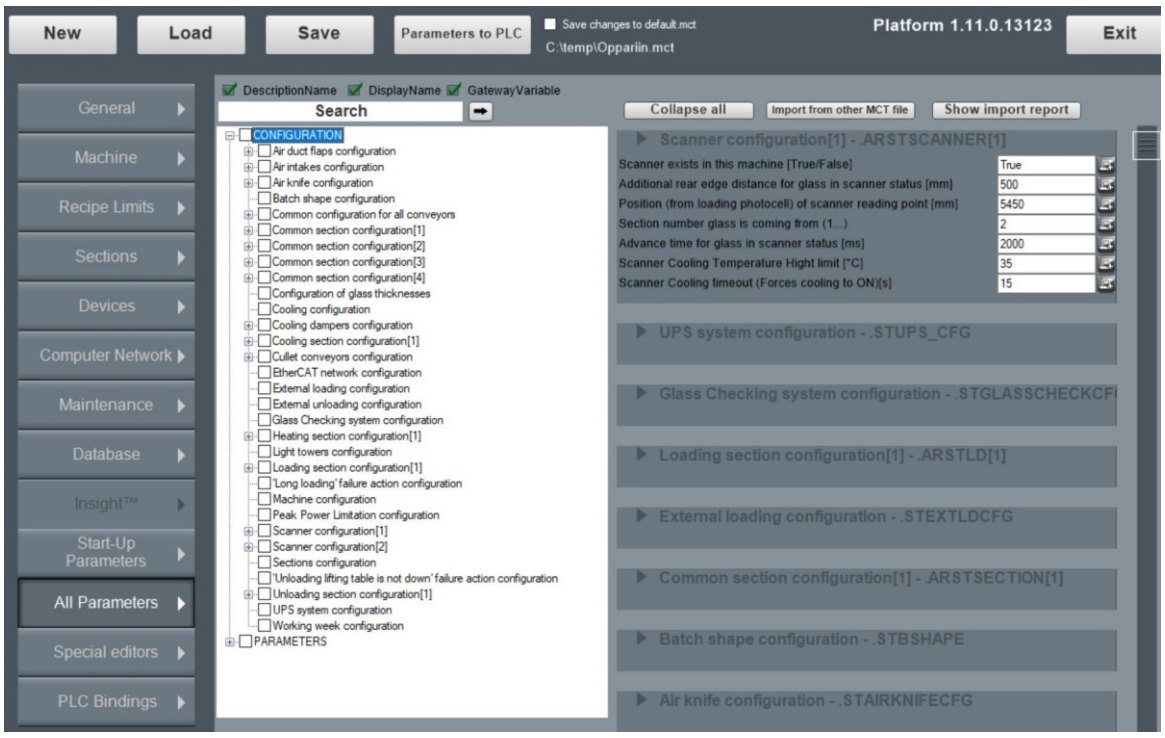
MCT-työkalussa on monta erilaista näkymää, jotka on jaettu välilehtiin. Editori aukeaa näkymään, jossa voi esimerkiksi määritellä käyttöliittymään asiakkaan nimen ja valita konetyypin ja käyttöliittymän kielivaihtoehdot. Välilehdillä on myös valintaruutuja, joita valitsemalla saa päälle tai pois jotain ominaisuuksia. Lisäksi editorin kautta voidaan päivittää tietokoneen asetustiedostoja. Jos Glastonin ylemmän tason järjestelmien tietokannat on kiinnitetty Microsoft SQL-serverin

kautta Glastonin tietokantoihin, niin MCT-tiedoston tallennus työkalulla jättää jäljen tietokannassa olevaan tallennuslokiin, josta muutoksia voi tarkistella jälkikäteen. Lokiin tallennetaan muutoksen tehnyt käyttäjä, aika, kommentit ja mitä on muutettu. Muuttuneita parametreja voidaan myös vertailla eri tallennuksien välillä ja tarvittaessa palauttaa jokin tietty versio.

5.1.3 Koneparametrien muokkaaminen nykyisellä MCT:llä

Koneiden parametrointiin käytetään tavallisesti mallisarjakohtaisia pohjatiedostoja, joissa on mallisarjakohtaiset perusparametrit tallennettuna. Pohjatiedostojen käytöllä saadaan parannettua konfigurointien samankaltaisuutta samanlaisten projektien välillä ja vähennettyä muutettavien parametrien määrää.

Pääasiassa koneparametrien muokkaaminen nykyisellä työkalulla tapahtuu työkalun ”All Parameters”- välilehdellä. Siellä alustavasti konetyypin mukaiseen pohjaan on valittu tietyt parametrirakenteet, joita muokkaamalla tiettyyn sarjaan kuuluvat peruskoneet saadaan konfiguroitua.

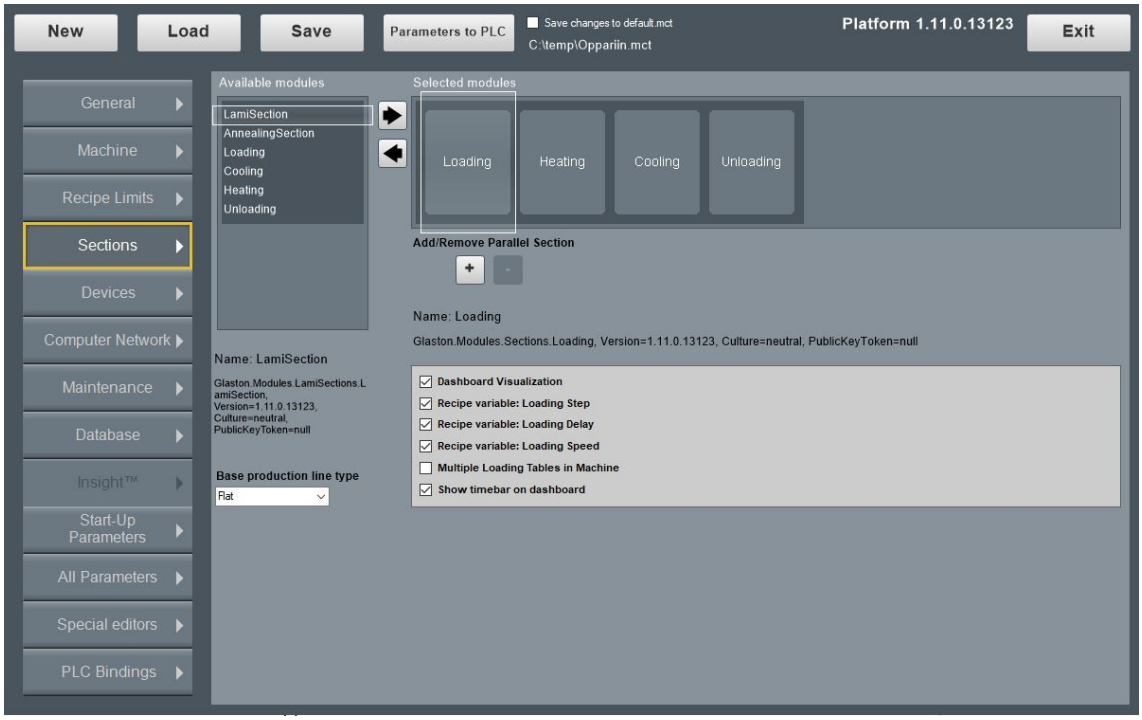


KUVA 8. MCT-työkalun parametrinäkymä

Parametrit on ryhmitelty näkymässä PLC-ohjelman rakenteen mukaisesti, joka tekee nykyisestä ryhmittelystä sekavan ja epäloogisen. Jotkut toimilaitteet ovat omana rakenteenaan ja jotkut taas jaoteltu osastojen mukaan. Sekava rakenne johtuu työkalun historian tuomasta painolastista, työkalua ei alun perin suunniteltu näin monen ominaisuuden ja parametrin muuttamiseen. Rakenteiden ja ominaisuuksien määrä on moninkertaistunut kymmenessä vuodessa, eikä työkalun toimintaperiaatteita ole muutettu tarpeiden mukaan.

5.1.4 Osastojen lisäys nykyisellä työkalulla

Noin puolessa parametroitavista koneista on neljä pääosastoa: lastausosasto, lämmitysosasto, jäähdytysosasto ja purkuosasto. Lopuissa koneista tarvitsee lisätä konfiguraatioon lisää osastoja, useimmiten toinen lämmitys ja toinen jäähdytysosasto kasvattamaan koneen tuotantokapasiteettia.



KUVA 9. Osastojen lisäys MCT-työkalulla

Alkuaikoina osastojen lisäys oli mahdollista, mutta nykyään ominaisuus ei enää toimi. Käytännössä lisäys tulee tehdä erillisellä XML-editorilla osaavan ja Glastonin käyttöliittymäpuolen tuntevan tekijän toimesta. Heitä on erittäin vähän ja ovat

täystyöllistettyjä. Työ vie aikaa ja on erittäin virheherkkää, pienikin virhe voi huonoimmassa tapauksessa aiheuttaa ongelmia asiakkaalla käyttöönotossa ja asiakasprojektien hyväksynnän viivästymisen, joka taas johtaa mahdollisiin rahallisiin kompensaatioihin.

5.2 Ohjelmistojen konfigurointi muualla

Ohjelmistojen konfigurointi on yleistä niin sanotuissa korkeamman tason ohjelmointikielissä. PLC-ohjelmoinnissa konfigurointi on harvinaisempaa, johtuen usein ohjelmien projektiluonteisuudesta. Muutokset ovat helpompi ja nopeampi tehdä suoraan lähdekoodiin, kun tehdä niistä konfiguroitavia ominaisuuksia myöhempää käyttöä varten. Usein projektille tehtävä työ on kertaluontoista, eikä samaa ominaisuutta tarvitsekaan enää myöhemmin. Projektiluonteisuuden takia PLC-ohjelman konfiguroinnista on vaikea löytää konfigurointiesimerkkejä. Glastonin tapa konfiguroida PLC-ohjelmaa ulkoisella tiedostolla on harvinainen, mutta toimivaksi todettu tapa, sekä mielestäni ”edellä aikaansa”.

5.2.1 Taajuusmuuntajien parametrointi

Yleisin esimerkki teollisuuden parametrointikohteista löytyy taajuusmuuntajien parametroinnista, jossa ohjaimen sisällä olevaa ohjelmistoa mukautetaan toimimaan vallitsevassa ympäristössä parametrien avulla. Parametroinnissa taajuusmuuntajalle kerrotaan usein esimerkiksi tietoja ohjaustavasta, moottorista, tavasta, jolla taajuusmuuntajan tulisi vastata ohjauspyyntöihin ja niin edelleen. Taajuusmuuntajien parametroinnin hankaluutena on usein ollut vaikeaselkoinen käyttöliittymä ja liian suuri määrä muutettavissa olevia parametreja.

Suurimpien taajuusmuuntajavalmistajien uusimmissa malleissa (esimerkiksi ABB) on myös alkanut yleistymään automaattisia ”Setup Wizard” -tyyppisiä ohjelmia, joilla parametrointiohjelma kysyy muutaman kriittisen kysymyksen, jolla saadaan määriteltyä ympäristötekijät oikein. Tämän jälkeen moottori pyöräyttää niin sanotun ”Autotune”-ohjelman, jolla taajuusmuuntaja tarkasta moottorista automaattisesti takaisinkytkennän avulla käyttäytymistietoja, jolla ohjelma viimeis-

telee määrittelyn. Tämän jälkeen moottorin tulisi olla valmis ajoon. Ohjattu käyttöönotto nopeuttaa ja selkiyttää moottorin käyttöönottoa, sekä vähentää virheiden määrää.

5.3 Setup Run -hanke

Syksyllä 2020 kehittämistehtävän aihetta mietittäessä esiin nousi yrityksessä samaan aikaan ideoinnissa ollut Setup Run -hanke. Hankkeen tarkoituksena on lyhyesti tehdä koneiden käyttöönotosta selkeä ja yksiselitteinen ”step-by-step” toimenpide, jolloin kaikki koneet asennetaan mahdollisimman samankaltaisesti riippumatta siittää, kuka asennusvalvojista koneen on ottanut asiakkaalle käyttöön. Tällä pyritään vähentämään asennusvalvojan käyttöönottokokemusvaatimusta, jolloin koneiden käyttöönottajien resurssointi helpottuu, koska käyttöönotto ei enää riipu asennusvalvojasta tai hänen osaamisestaan. Eräs tuotekehityksen työntekijä kiteyttikin tilanteen hyvin seuraavasti:

> Ongelmatilanteissa ja vianmäärityksissä selvittäminen helpottuu, jos koneet asennetaan samalla tavalla. Asentajakohtaiset ”muuttuvat tekijät” pyritään jättämään pois, jolloin seuraamalla määriteltyjä vaiheita asennus muuttuu rutiininomaisemmaksi. Tämä nopeutta asennusta ja selkeyttää resursointia. Joillekin projekteille, joihin asiakas on ostanut nopean asennus- ja käyttöönottopalvelun, voidaan siten resursoida useampi käyttöönottaja ja jakaa työtehtävät etukäteen tehtävälistan mukaisesti.

Koneiden käyttöönoton yhtenäistämisen tarkoituksena on saada tuotantoparametrien säätö yhtenäiseksi esimerkiksi kahden samanlaisen, mutta eri käyttöönottajan toimesta eri paikkoihin asennettujen koneiden välille. Samalla periaatteella, jos Glaston myy äskeisen esimerkissä mainittujen kahden koneen lisäksi kolmannen samanlaisen koneen ja se asennetaan aikaisempien koneiden kanssa identtisesti, voidaan asiakkaalle myydä vastaavissa koneissa hyvin toimineet tuotantoparametrit, jolloin asiakas voi aloittaa suoraan huippulaatuisen tuotannon, eikä konekohtaisia säätöjä tarvitse hioa viikkotolkulla. Sama pätee myös, jos joskus myöhemmin asiakas haluaa ostaa aiemmin Setup Runin mukaan asennettuun koneeseen valmiit muualla vastaavassa koneessa hyvin toimineet

tuotantoparametrit niin hän saa koneensa nopeasti tuotantokelpoiseksi haastavillekin prosesseille. Haastatteluissa tuotekehityksen työntekijä toteaakin tilanteesta seuraavaa:

> Yleensä asiakkailla kestää prosessin hiomisessa uudelle lasityypille viikkoja, huippulaatuun päästäkseen säätöjä voi joutua tekemään kuukausia. Lisäksi osaajat, jotka säätävät asiakkaalla uutta prosessia ovat haluttuja ja kalliita, joten pienemmissä yrityksissä osaaminen johonkin tietyntyyppiseen prosessiin olisi huomattavasti nopeampi ja jopa halvempi ostaa esimerkiksi Glastonilta kuin palkata ammattilainen fyysisesti paikalle huolehtimaan prosessin säätämisestä.

Esimerkkinä tästä: Jos asiakas voisi ostaa tuotantoparametrejä, he voisivat ottaa prosessoitavaksi semmoisia tuotteita, joista heillä ei ole kokemusta, mutta joita heidän asiakkaansa haluaa prosessoitavan ja on valmis maksamaan siitä hyvän hinnan. Näin Glaston hyötyisi tietotaidostaan ja asiakas tuotannon joustavuudesta, lisäksi asiakkaan asiakas saisi nopeasti laadukkaasti prosessoidun tuotteen.

5.4 Suunnittelun osuus hankkeessa

Vaikka Setup Run -hanke alun perin liittyikin vain koneiden käyttöönoton kehittämiseen, haluttiin yrityksessä viedä käytäntöjen standardointi myös koneiden suunnitteluun. Jokainen yrityksen sisäinen suunnitteluosasto, mekaniikka-, sähkö- ja automaatiosuunnittelu suunnittelevat ja kehittävät toimintojensa yhtenäistämistä. Tavoitteena myös suunnittelupuolella on käytäntöjen yhtenäistäminen niin, että suunnittelun lopputuloksen laatu ei ole suunnittelijasta tai hänen kokemuksestaan riippuvaista. Jos esimerkiksi automaatiosuunnittelussa kaksi samanlaista konetta esikonfiguroidaan kahden suunnittelijan toimesta, joista toinen on ollut töissä vuoden ja toinen 15 vuotta, koneet saattavat toimia eri lailla, vaikka se olisi lopulta asennettu Setup Run -ohjelman kautta samalla tavalla. Samalla tavoitellaan myös suunnittelun laadun parantamista ja projektin suunnittelun läpivientiajan nopeuttamista.

Kuten jo aiemmin nykyisten toimintatapojen esittelyssä mainittiin, vakioprojektien automaatiosuunnittelu voidaan jakaa kahteen osa-alueeseen, I/O-konfigurointiin ja koneparametritiedoston (MCT-tiedoston) konfigurointiin. Näistä ensimmäisenä

mainittu I/O-konfiguroinnin kehittäminen Setup Run- hanketta varten etenee omana projektinaan, jossa nykyisin hyvin vaivalloisesti ja hitaasti muokattava I/O-konfigurointi voidaan tehdä jatkossa niin, että projektille suunniteltu I/O-konfiguraatio voidaan tuoda suoraan sähkökuvista XML-tiedostona ja lisätä I/O-konfigurointi-ohjelmaan. Näin vältytään inhimillisiltä virheiltä I/O-konfigurointia tehdessä ja homma nopeutuu merkittävästi. Varsinaisen koneen konfigurointitiedoston konfiguroinnin kehittäminen taas toimii tämän opinnäytetyön aiheena.

5.5 Kehittämisen aloitus

Kun MCT-konfiguroinnin kehittämistä alettiin ideoimaan syksyllä 2020, tarkoitus oli tehdä kokonaan uudenlainen tyyli konfiguroida koneen MCT-tiedosto uudentyyppisellä ”step-by-step” työkalulla, jota käytetään myös koneen käyttöönottoon asiakkaalle. Alkuvuodesta 2021 kävi kuitenkin ilmi, että yrityksen resurssit ei riitä ainakaan tässä vaiheessa kokonaan uudentyyppisen konfiguroinnin kehittämiseen siinä laajuudessa, mitä oli alun perin suunniteltu. Koska projekti oli kuitenkin jo aloitettu, päätettiin kehittämisen suunnittelu jakaa kolmeen eri vaiheeseen.

Ensimmäinen kehitysvaihe kerää yhteen ne parametrit, joita yleensä muokataan normaalia konetta alustettaessa. Koneen peruskonfiguroinnin kannalta epäolennaiset parametrit on jätetty kokonaan pois sekaannusten välttämiseksi. Nämä parametrit tehdään muokattavaksi Setup Run -työkaluun. Jos koneessa on jotain erikoista, pitää parametrointi suorittaa vanhalla työkalulla, koska uudella ei ominaisuuksien tai rakenteiden lisäys onnistu.

Toinen vaihe tuo ensimmäiseen lisäyksenä valmiiden parametrirakenteiden tuontimahdollisuuden työkaluun ulkoisesta lähteestä, esimerkiksi verkkolevyllä Excelistä. Nämä rakenteet ovat usein virheherkkiä parametrikokonaisuuksia, jotka ovat fiksuinta ladata konfiguraatioon kokonaisuutena.

Kolmas vaihe on oikeastaan ”täydellinen konfigurointityökalu” -tutkielma. Se tuo ensimmäiseen ja toiseen vaiheeseen lisäyksenä uudentyyppisen parametrointinäkymän ja tekee konfiguroinnista vaiheistettua ja selkeää. Tavoitteena on tehdä konfiguroinnista yhtä helppoa kuin ohjelmien asennuksesta tietokoneelle.

Vaiheet on jaoteltu niin, että ensimmäinen vaihe tullaan toteuttamaan Setup Run-projektin alkuvaiheessa, koska se ei sisällä muuta kuin nyt useimmin muokattujen parametrien valikoimisen uuden työkalun näkymään semmoisenaan ja kaikki turha jätetään pois. Toisen vaiheen toteuttamisajankohtaa ei tiedetä, mutta ainakin parametrien automatisointeja tullaa tuomaan Setup Run- työkaluun mahdollisesti seuraavana, mutta tuskin tämän opinnäytetyön kirjoitusaikana. Toisen vaiheen muut ominaisuudet ja kolmannen vaiheen ideat tehdään kehittämisehdotuksena, jos työkalua halutaan kehittää vielä pidemmälle.

6 ENSIMMÄINEN VAIHE

Kuten edellisessä kappaleessa mainittiin, ensimmäinen vaihe tullaan toteuttamaan Setup Run -työkalun alkuvaiheessa, jolloin osa MCT-tiedoston nykyisestä konfiguroinnista suoritetaan uudella työkalulla. Jako uuden ja vanhan työkalun käytön suhteen tullaan tekemään siten, että nykyisellä MCT-työkalulla tullaan tekemään koneen käyttöliittymän konfigurointi ja PLC-parametrit siirrytään muokkaamaan uuden työkalun näkymään. Uuden työkalun näkymään on karsittu esillä olevia PLC-parametreja niin, että muokattavissa on vain ne parametrit, joita tyypillisen koneen konfiguraatioon yleensä muutetaan. Jos koneessa on jotain tyypillisestä konekonfiguraatiosta poikkeavaa ja piilossa olevia parametreja jouduttaisiin muokkaamaan, muokkaus tehtäisiin vanhalla MCT-työkalun näkymällä tai XML-editorilla.

> Ensimmäisen vaiheen tarkoituksena on päästä sisään työkalun käyttöön ja tuoda se rutiiniksi suunnittelijoille, vaikka se joissain tapauksissa lisäisikin konfiguroinnin monimutkaisuutta, kun ohjelmia joudutaan kesken konfiguroinnin vaihtamaan (Automaatiosuunnittelun päällikkö 2021).

Varsinkin kokemattomimmille suunnittelijoille, esimerkiksi kesätyöntekijöille uuden työkalun näkymä tuo paljon lisäarvoa, kun kaikkea olemassa olevaa ei pääse säätämään, koska näkyvillä on uudessa työkalussa vain oleelliset parametrit. Lisäksi ohjeet uuden työkalun PLC-parametrien konfigurointiin on huomattavasti helpompi tehdä, kun valinnanvaraa ei ole paljoa.

Osalle lukijoista herää varmasti tässä vaiheessa konfiguroinnin suunnittelua kysymys, miksi sitten ei jäätäisi tähän? Jos kerran ensimmäisen vaiheen kriittisten parametrien muokkaus onnistuu uudella työkalulla pienellä vaivalla, miksi työkalua tulisi kehittää eteenpäin? Syy jatkokehitykseen on koneisiin myytävät lisäoptiot, räätälöidyt ratkaisut ja erikoiskoneet.

> *Noin puolet myytävistä uusista koneista on helppoja ja nopeita konfiguroida, eivätkä ne luo tarvetta kehittää työkalua. Erikoiskoneiden konfigurointi vie merkittävästi enemmän aikaa kuin tavallisten koneiden konfigurointi, lisäksi erikoiskoneiden konfiguroinneissa sattuu*

> *eniten virheitä muokattavien parametrien suuren määrän takia.* (Automaatiosuunnittelun päällikkö 2021)

Erikoiskoneiden konfiguroinnin kehittäminen ja laadun parantaminen onkin suuri motivaattori jatkaa työkalun kehittämistä.

7 TOINEN VAIHE

7.1 Parametrien hallinnointi

Setup Run -työkalun käytön toiseen vaiheeseen tuotaisiin muutamia parametrointia helpottavia ominaisuuksia. Projektialustuksessa käytettävien pohjakonfiguraation parametrointia kehitetään hyvin vähän ja ainoastaan selvissä ongelmatilanteissa. Automaatiosuunnittelun päällikkö toteaakin haastatteluissa seuraavaa:

> Pohjakonfiguraatioiden parametrien muokkaaminen tapahtuu pääsääntöisesti tuotehallinnan pyyntönä suunnittelijalle muokata jotain tiettyä ohjelmaparametria jossain konesarjassa tai joskus jopa kaikissa. Korjauspyynnöt ovat harvassa ja korjattavien parametrien korjaustarve tulee usein esille jonkun toisen asian kautta puolivahingossa, kun kone ei asiakkaalla käyttäydykään, kuten prosessista vastaava erityisasiantuntija on ajatellut. Monesti tuotehallinnalla ei ole juurikaan tiedossa, mitä prosessin kannalta kriittisiä asetuksia meillä koneisiin laitetaan, vaikka he vastaavat tuotantoprosessin toimivuudesta ja laadusta.

Ensimmäinen suuri muutos tulee olemaan koneenalustusparametrien hallinnoinnin ja muokkauksen siirtäminen tuotehallinnalle. Tuotehallinta yhdessä tuotekehityksen kanssa ylläpitää koneen toimintaan vaikuttavia parametrejä, joten tuotteen laadusta ja toimivuudesta vastaava taho on siten itse vastuussa parametrien ylläpidossa, eikä yksittäinen usein pelkästään tekniseen toteutukseen keskittyvä suunnittelija.

Koneenalustusparametrien hallinnointi on nykyisin osana koneenalustus XML-tiedostoa, jota on vaikea lukea ilman konfigurointiin tarkoitettua MCT-työkalua. Koneen ohjausparametrit tulisi säilyttää selkeämmin luettavassa muodossa, joka osaisi lukea ja jäsentää parametreja sisältävän XML-tiedoston helposti luettavaan ja muiden pohjaprojektien kanssa vertailtavaan muotoon. Näin tuotehallinnan olisi helppo lukea koneen parametreja ja tehdä mahdollisia muokkauksia. Tiedostojen muutoshistoria tulisi myös olla luettavissa jälkikäteen, kuka on muokannut mitäkin pohjaparametria ja miksi.

Kun suunnittelija alkaa työstämään projektiaan, hän lataa verkosta ylläpidetyn pohjan, johon hän tekee tarvittavia koneen konstruktioon liittyviä muokkauksia. Jos suunnittelija tarvitsee jotain tiettyjä ominaisuuksia konfiguraatioonsa (esimerkiksi myytyjä lisäominaisuuksia) joita peruspohjassa ei ole, hän voi ladata toiminnon kokonaisen parametrirakenteen semmoisenaan ylläpidetystä pohjasta, jonka hän tuo senhetkiseen konfiguraatioonsa.

7.2 Parametrien sisäiset linkitykset ja niiden riippuvuudet

Kun tutkii tarkemmin nykyistä konfiguraatiota, huomaa että projektialustuksessa on paljon parametreja, jotka ovat ”linkitetty” keskenään ja siten riippuvaisia toisistaan. Tällaisten parametrien kanssa sattuu usein huolimattomuusvirheitä ja unohduksia, vaikka parametrejä voitaisiin laskea aiemmin muualla ilmoitetuista parametreistä. Keskusteluissa käy selväksi, että toisessa vaiheessa näihin ennaltaarvattaviin parametreihin haluttaisiin ”automaattitäyttö”, jonkun kriittisen parametrin täyttämisen jälkeen parametrointi-ohjelma ehdottaa siitä täytetystä parametrista riippuvien parametrien automaattitäyttöä, joka vähentäisi huolimattomuusvirheitä ja nopeuttaisi konfigurointia.

Otetaan tästä esimerkkinä koneella ajettavan lastauksen enimmäispituus, josta automaatiosuunnittelun päällikkö toteaa seuraavaa:

> Pelkästään tämän parametrin syöttämisellä voitaisiin määritellä koneelle esimerkiksi oletettavat osastojen pituudet, oskillointi-alueiden mitat, konvektion aloitus ja lopetuspaikat ja niin edelleen. Nyt koneen lastauksen enimmäispituuteen liittyviä parametrimuutoksia tehdään useita kymmeniä jokaiseen koneeseen yksitellen laskemalla, vaikka ne voitaisiin laskea etukäteen yhdestä ainoasta syötetystä mittatiedosta. Silloin tällöin joku parametreista unohtuu täyttää projektin alustajalta, jonka seurauksena kone saattaa toimia miten sattuu asiakkaalla.

Alkuvaiheessa tällaisten kriittisten sidosparametrien syöttämisen jälkeen konfigurointityökalun tulisi ehdottaa suunnittelijalle ennalta laskettuja suositusmuutoksia, jotka käyttäjä voi hyväksyä tai hylätä. Myöhemmin jos toiminto todetaan toimivaksi, konfiguraatio voi itse tehdä muokkauksia ja käyttäjä voi nähdä halutessaan yhteenvedon konfigurointiohjelman itsenäisesti muokkaamista parametreista.

7.3 PLC parametrirakenteiden päivitys

Konfigurointiparametrirakenteita tulisi miettiä PLC ohjelmassa uudestaan. Osa ohjelmassa olevista rakenteista on määritelty spesifisten osastojen alle (lastaus, lämmitys, jäähdytys ja purku) ja osa taas laitettu rakenteellisesti ”johonkin” ilman tarkempaa suunnitelmaa. Uudet lisättävät ominaisuudet ovat laitettu viimevuosina usein ”juuritasolle”, joka on ollut helpoin paikka uudelle parametrirakenteelle moniosastoisien koneiden määrän jatkuvan kasvun takia. Tämä aiheuttaa ongelmia parametrointinäkymässä, koska ominaisuuksia ei ole aina osastoittain, vaan niitä on ”siellä täällä”. Ominaisuuksien lajittelu osastoihin selkeyttäisi konfigurointia.

8 KOLMAS VAIHE

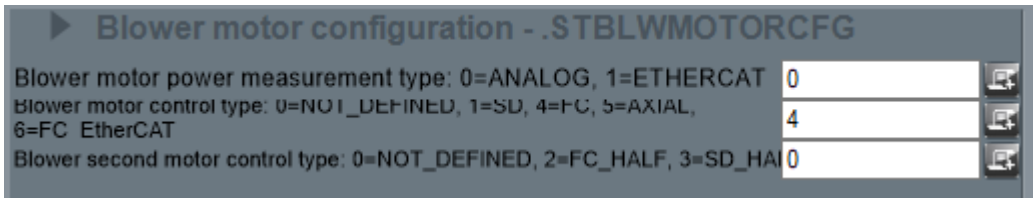
8.1 Uudistettu parametrointi

Kolmannen vaiheen parametrointi tuo koneiden konfigurointiin merkittäviä muutoksia. Kahdessa ensimmäisessä vaiheessa käytettyjä ”pohjatiedostoja” ei enää tarvita, vaan parametrointitiedosto luodaan tyhjästä työkalun avulla. Ohjelma kyselee tarvittavia kysymyksiä, esimerkiksi koneen mallisarjan ja konetyypin, enimmäispituudet ja leveydet ja valittuun konetyyppiin valitut optiot. Ohjelma hakee verkon kautta tuotehallinnan määrittelemät mallisarjakohtaiset vakio-ominaisuudet ja näyttää myös esimerkiksi tiettyyn konekokoon saatavilla olevat optiot. Kun koneen optiot on valittu, ohjelma hakee verkosta määritellyn koneen mukaiset rakenteet ja generoi alustavan parametrointitiedoston.

Tämän jälkeen ohjelma alkaa kyselemään osastojen parametrointiin liittyviä ”ydinkysymyksiä” osasto kerrallaan lastausosastosta purkuosastoon. Kysymysten vieressä tulisi olla jonkinlainen lisätietopainike (esimerkiksi ”?”-nappi), jonka takaa löytyy tarkempi parametrointiohje kyseiselle parametrille. Vaikeampien kysymysten kohdalla voitaisiin kysyttävä asia havainnollistaa konfiguroijalle visuaalisemmin esimerkiksi kuvalla tai lyhyen animaation avulla. Parametroinnin toisen vaiheen kohdassa 7.2 esitellyt parametrien väliset sidokset säilyvät myös kolmannessa vaiheessa, jolloin ohjelma ehdottaa muutoksia muihin parametreihin, jotka ovat riippuvaisia käyttäjän juuri muokkaamasta parametrista. Tarpeen tullen suunnittelijoiden tulisi päästä muokkaamaan parametrirakenteita myös vapaasti, mutta tämä tapahtuisi vain erikoistapauksissa ja vain tietyt oikeudet omaavan henkilön toimesta. Keskusteluissa asetetaan kolmannelle vaiheelle selkeä tavoite: Työkalu tulisi suunnitella niin, että sen avulla on mahdollista parametroida 90 % myytävistä tuotteista ilman, että tarvittaisiin ”harvoin muokattavien” parametrien muokkausta.

Kysymysten esittämistä ja asettelua tulisi miettiä uudestaan. Monet PLC-rakenteeseen tehdyt muuttujien ”selitykset” ovat välillä todella kryptisiä ja harhaanjohtavia. Tämä johtunee nykyisen työkalun suhteellisen lyhyestä selityskentästä, jo-

hon työkalu kopioi suoraan muuttujan jälkeiset kommentit PLC rakenteesta. Monet muuttujat sisältävät monta vaihtoehtoa esimerkiksi jonkun toimilaitteen ohjaustavalle, kuten alla olevassa kuvassa (KUVA 10). Alemmassa kuvassa (KUVA 11) on ylemmän kuvan (KUVA 10) mukainen parametrirakenne PLC-puolella, josta se tuodaan MCT-työkaluun semmoisenaan. Nykyinen työkalu yrittää skaalata tekstiä pieneen kenttään leikaten osan ensimmäisen rivin tekstistä pois, joten lopputulos ei ole kovin luontevan näköinen.



KUVA 10. MCT-työkalun kompaktit selityskentät

```
(* Enums... *)
eBlwMotorPwrMeasType: E_BLW_MTR_PWRMEAS_TYPE;    (* Blower motor power measurement type: 0=ANALOG, 1=ETHERCAT *)
eBlwMotorType:        E_BLW_MTR_MODE;            (* Blower motor control type: 0=NOT_DEFINED, 1=SD, 4=FC, 5=AXIAL, 6=FC_EtherCAT *)
eBlwSecMotorType:     E_BLW_MTR_MODE;            (* Blower second motor control type: 0=NOT_DEFINED, 2=FC_HALF, 3=SD_HALF *)
```

KUVA 11. Konfigurointiparametrin näkymä PLC-puolella

8.2 Konfiguraatio sähkökuvista

Tapaamisissa selviää myös, että projektit voitaisiin konfiguroida ainakin osittain myös projektin sähkökuvista saatavan erillisen koneen tulo- ja lähtötietojen (I/O) perusteella:

> *Yrityksessä lähitulevaisuudessa sähkökuvien suunnittelussa käyttöönotettavan EPLAN Electric P8- ohjelmasta on mahdollista saada tulo- ja lähtötiedot osoitteineen XML-formaatissa, jota kyetään jo nyt hyödyntämään automaatiosuunnittelussa niin, että kenttäväyläkonfigurointi voidaan rakentaa tämän tiedoston perusteella.*

Glastonilla on tätä varten oma työkalu, joka yhdistää I/O-pisteen osoitteen ja tälle pisteelle määritellyn ohjelmamuuttujan, jonka jälkeen PLC-ohjelma osaa käyttää

sitä I/O- pisteiden luennassa kenttäväylästä. Näin normaalisti muutamasta tunnista useisiin kuukausiin kestävä I/O-projektin rakentaminen projektille saadaan automatisoitua lähes täysin.

Samaa I/O-projektin XML-tiedostoa voitaisiin käyttää hyödyksi myös laitekuvauskonfiguraation tekemisessä. Sen kautta voitaisiin määritellä yksinkertaisia tietoja, esimerkiksi montako tulotietoa jollain prosessin toimilaitteella on sähkökuviin piirretty linkattujen muuttujamäärien perusteella tai että onko projektille lisätty joku erillinen optiona olosuhteiden takia menevä toimilaite (esimerkkinä UPS, joka on akkuvarmennus vaihtovirtapuolelle sähkökatkoja varten). Usein nämä vähemmän merkittävät asiat jäävät usein huomiotta konfiguraatiota tehdessä, jos niihin on tullut muutoksia. Konfigurointiohjelma voisi tiedoston luennan jälkeen antaa käyttäjälle mahdollisuuden yksi kerrallaan hyväksyä muutokset tai loppuyhteenvedon tehdyistä muutoksista. Käyttäjällä tulisi olla mahdollisuus valita kumpaa tapaa hän haluaa tilanteen mukaan käyttää.

8.3 Alustusparametrien määrittely ja hallinta

Koneita konfiguroivan automaatiotiimin ja tuotehallinnan välillä on jo pitkään ollut epäselvyyttä, kuka hallitsee konfiguroitavien koneiden alustusparametreja. Tuotehallinta vastaa tuotteesta ja määrittelee, kuinka tuotteen kuuluisi toimia. Tuotehallinta määrittelee yhdessä tuotekehityksen kanssa asiakkailla testattujen koneenohjausparametrien määrittelystä pohjatiedostoihin, joilla pyritään saavuttamaan mahdollisimman hyvä laatu tuotteelle. Monesti nämä parametrit luovat pohjaa prosessinohjaukselle ja kuinka asiakkaan muokkaamat tuotantoparametrit käyttäytyvät prosessissa.

Keskusteluissa selviää, että tuotehallinnalla ei kuitenkaan ole työkaluja nähdä, kuinka koneet on konfiguroitu, koska tämä näkyvyys on ollut vain automaatiotiimin ohjelman kautta automaatiotiimillä:

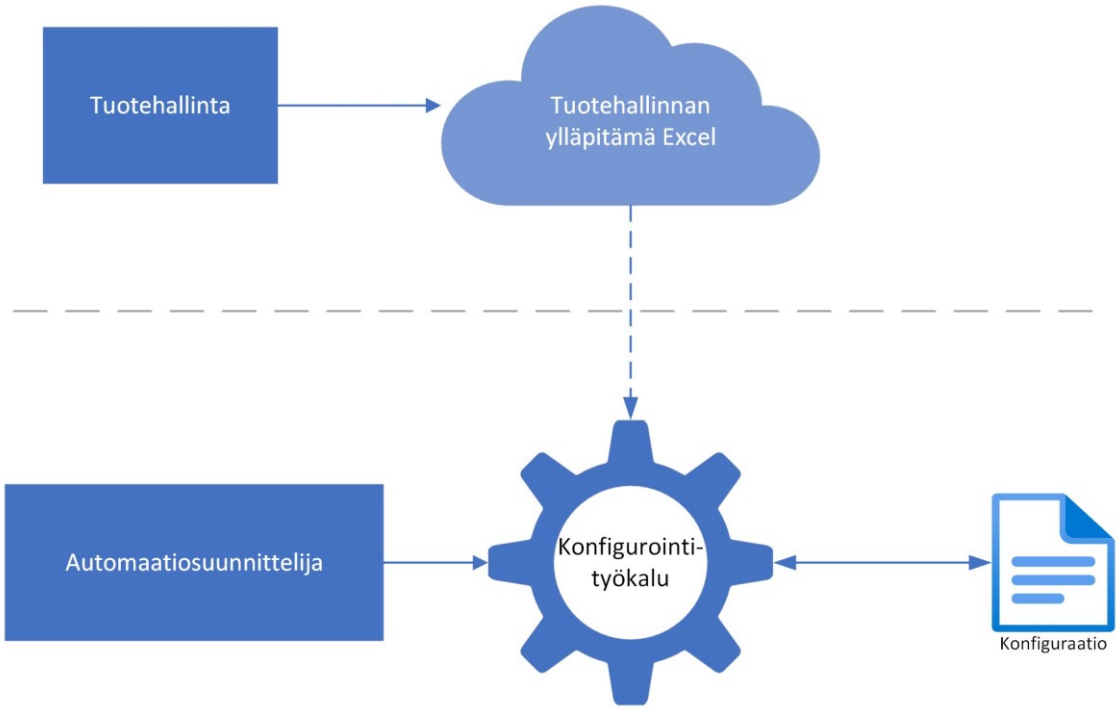
> Tämä on jo pitkään aiheuttanut ongelmia, kun tuotehallinta on luullut lähtevien tuotantokoneiden olevan tehty tietyllä tavalla, vaikka totuus voi olla aivan toinen. Jos projekteilla on tullut laadun kanssa ongelmia, tuotehallinta on yhdessä tuotekehityksen kanssa pohtinut ratkaisuja ongelman selvittämiseksi, mutta ongelma ei ole ratkennutkaan ”näkymättömien” prosessiin vaikuttavien pohjaparametrien takia.

Tästä syystä tuotetta hallitsevilla toimijoilla, tässä tapauksessa juuri tuotehallinnalla ja tuotekehityksellä, tulisi olla hyvä näkyvyys näihin tuotteen prosessiin vaikuttaviin perusparametreihin, joita he pääsisivät muokkaamaan asiakkailta saadun datan avulla. Hehän ovat lopulta vastuussa, että asiakkaat voivat Glastonin koneilla tuottaa laadukasta lopputuotetta.

Konfigurointitiedostosta on todella työlästä tarkistaa arvoja rakenteellisuutensa takia ja nykyinen konfigurointityökalu on myös aikamoinen parametriviidakko. Työkalun ylläpito on osoittautunut raskaaksi, eikä yrityksellä ole riittävästi resursseja sen käyttämiseen ja jatkokehittämiseen tuotehallintaa varten. Tämän takia yksi tapa esitellä visuaalisesti tuotantoparametrikokonaisuuksia olisi Excel. Exceliin voitaisiin tehdä konekokokohtaisia rakennekokonaisuuksia, joita voitaisiin lukea parametrointivaiheessa esimerkiksi verkon yli verkkokansiossa olevasta Excel-tiedostosta. Excelin käyttö parametrien määrittelyssä madaltaa myös selvästi kynnystä määrittelevällä taholla tarkastella konekokokohtaisia arvoja. Lisäksi Excelin kautta pystytään luomaan nopeita visualisointeja, joilla voidaan tarkastella joitain tiettyjä ominaisuuksia taulukoista. Jos haluaa mennä visualisoinneissa vielä pidemmälle, voidaan esimerkiksi Exceliin tehtyjä taulukoita viedä PowerBi-työkaluun ja yhdistellä sitä asiakkailta saatuun tuotantodataan. Tämä voisi auttaa havainnoimaan paremmin laatuhaasteista kärsivien projektien alustusparametreja vertailemalla niitä hyvin toimivan koneen alustusparametreihin.

8.3.1 Parametrien lataus Excelistä konfiguraatioon

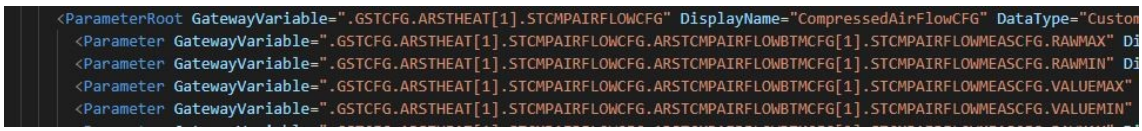
Excelissä olevia parametrirakenteita tulisi olla kattavasti eri konevariaatiolle. Excelistä haettaessa tietyn koneen tietyntyyppisiä parametrirakenteita työkalun tulisi pystyä kertomaan hakutapahtumassa minkälainen kone on kyseessä. Jos vaihtoehtoja on useampi, käyttäjälle annetaan valittavaksi, kumpi mahdollisista konfiguraatioista valitaan havainnollistamalla niiden ero. Ihannetilanteessa käyttäjän ei tulisi joutua valitsemaan, vaan oikein syötettyjen lähtötietojen perusteella tulisi löytyä ainoastaan yksi vaihtoehto, joka sitten ladataan konfiguraatioon.



KUVA 12. Parametrien lataus tuotehallinnan ylläpitämästä Excel-tiedostosta.

Excelissä olevien taulukoiden latausta konfiguraatioon käytetään jo osittain nykyisessä MCT-työkalussa, joten teknologia tähän on jo olemassa. Kehittämällä jo olemassa olevaa tekniikkaa eteenpäin voitaisiin tuoda suurempia kokonaisuuksia alustusparametrien rakenteisiin ja pitää samalla rakenteet helpommin luettavassa muodossa.

Suurin ongelma Excelin käytössä parametriarvojen säilytyspaikkana on parametrien polkujen pitäminen toimivana, jotta parametriarvon lataus onnistuu oikean välilehden oikeasta solusta. Parametrejä lisättäessä rakenteeseen rakenne menee äkkiä ”rikki”, jolloin parametria ladattaessa saadaan Excelistä haun tuloksena väärän parametrin arvo. Tästä syystä arvojen lisäksi rakenteiden polut tulee kopioida Excel-välilehdelle arvon yhteyteen, jolloin rakenteita ladattaessa parametri ja sen arvo kulkevat yhdessä, jolloin arvojen sekoittumisen riski pienenee huomattavasti. Tämä polku on käytännössä PLC:n konfigurointirakenteen mukainen polku tietylle parametrille, jonka avulla CONN-serveri kirjoittaa ADS-rajapinnan yli PLC parametrirakenteeseen oikean arvon oikeaan paikkaan.



KUVA 13. Parametrirakenteen polku on tarkkaan määritelty

9 JOHTOPÄÄTÖKSET JA POHDINTA

9.1 Opittua

Tutkimusprojekti oli mielenkiintoinen syväluotaus tavallaan ennestään tuttuun konfiguroinnin maailmaan, jossa osa ongelmista oli jo itselle hyvin tuttuja. Varsinkin käsite ”massaräätälöinti” (sivulla 14) oli erittäin mielenkiintoinen kokonaisuus alakäsitteineen. Työssä yllätti tiedon löytämisen hankaluus internetistä eri hakukoneiden kuin digikirjallisuudenkin osalta. Yleistason käsitteitä löytyi, mutta muuten varsinaisten toteutusten löytyminen esimerkein oli erittäin hankalaa. Kuten kirjallisuuskappaleen yhteenvedossa mainitsinkin, suorat esimerkit ovat hyvin todennäköisesti liikesalaisuuksia, joita ei haluta yleisesti jaettavaksi.

Tutkimuksen aikana pääsi kartoittamaan vaihtoehtoisia tapoja tehdä konfigurointia. Mieleisin homma oli ehdottomasti kolmannen vaiheen (kappale 8) visiointi, jossa pääsi keksimään puhtaalta pöydältä täysin uudenlaisia lähestymistapoja konfigurointiin. Työtä tehdessä huomasi nopeasti, että mitä ”automaattisemmaksi” konfiguroinnin pystyy tekemään, sen parempi ja nopeampi se on. Automaattisuus tuo toki muita haasteita, esimerkiksi sen, että automaattisuus tulee olla hyvin hallinnassa ja läpinäkyvää, jotta ”vääriä muutoksia” saadaan konfiguraatioista nopeasti pois.

9.2 Vastaaminen tutkimuskysymyksiin

Vastauksia tutkimusten pohjalta alussa esitettyihin tutkimuskysymyksiin:

1. Mitkä ovat koneen konfiguroinnin kannalta kriittisiä parametreja?

   Koneen konfiguroinnin kannalta kriittiset parametrit käytiin läpi automaatiosuunnittelun päällikön ja myöhemmin myös muiden automaatiosuunnittelijoiden kanssa erillisessä palaverissa, jonka pohjalta niistä tehtiin lista. Lista otettiin itseasiassa heti käyttöön ”tarkastuslistaksi”, josta voi tarkistaa, että kaikki yleisemmin muutosta vaativat parametrit on käyty läpi.

2. Kuinka kriittisten parametrien muokkausnäkymä esitellään suunnittelijalle?

   Konfiguroinnin muokkausnäkymän esittely suunnittelijalle muuttuu vaiheittain ja lopuksi kolmannessa vaiheessa typistyy vain ydinkysymysten kyselyyn. Kolmannessa vaiheessa suunnittelija näkee hyvin vähän parametreja parametroinnin aikana, joka nopeuttaa työtä ja vähentää inhimillisiä virheitä.

3. Tärkeiden parametrien sidossuhteet muihin parametreihin ja niiden muokkauksen automatisointi ohjelmalla?

   Tärkeiden parametrien sidossuhteita selviteltiin projektin alussa ja ne jaettiin käytännössä kolmeen eri ryhmään: sidossuhteettomat parametrit (parametri ei ole sidoksissa mihinkään muuhun parametriin), sidossuhteelliset parametrit (parametri on sidoksissa johonkin toiseen parametriin) ja ehdollisesti sidossuhteelliset parametrit (parametri on vain joissain tapauksissa sidoksissa toiseen parametriin, esimerkiksi option yhteydessä).

4. Kuinka uusien parametrien tuodaan ohjelmaan?

   Uusien parametrien tuonti ohjelmaan ja olemassa oleviin rakenteisiin on edelleen haastavaa. Parametrien lisäys tai muuttaminen olemassa oleviin rakenteisiin on työlästä ja virhe herkkää. Lisäksi parametrin lisää usein automaatiosuunnittelija, mutta lisäys tehdään versionhallinnallisesti eri paikkaan, kun muu automaatiosuunnittelu.

5. Kuinka konfiguroinnista ohjelmalla saa tehtyä niin helppoa, että konfiguroinnin voisi tarvittaessa suorittaa joku muu kuin automaatiosuunnittelija?

   Konfiguroinnin helpottamiseen mallia voidaan ottaa esimerkiksi taajuusmuuntajien käyttöönottoon tarkoitetun erillisen asennusohjelman käyttölogiikasta, jossa ennakkoon tarkkaan määritellyt kysymykset kysytään tietyssä järjestyksessä konfiguroijalta. Ohjelma itse tekee varsinaista konfigurointia taustalla automaattisesti, mutta oletettavasti aina samalla lailla. Ohjelman lopputuotteena tulisi olla suunnittelijan taidoista riippumaton laadukas lopputulos.

9.3 Johtopäätökset ja yhteenveto

Nykyinen tapa konfiguroida koneita toimii, kunhan tekijä on huolellinen ja hänellä on kokemusta konfiguroinnista. Hänen tulee myös saada takaisinkytkentää konfigurointeihinsa asiakkaalla koneissa havaituista konfigurointivirheistä, jotta seuraavat alustukset menisivät paremmin. Tekijän tulee olla valveutunut koneen uusista ominaisuuksista ja kuinka ne konfiguroidaan päälle ja missä olosuhteissa. Tekijä alkaa kerryttämään sisälleen ”hiljaista tietoa” koneen konfiguroinneista, josta kaikkea on vaikea dokumentoida. Jos tekijöitä olisi yrityksessä vain yksi, alkaa hän tekemään itsestään ”korvaamatonta”.

Tästä syystä konfigurointi tulisi saada tehtyä niin helpoksi, että kuka vain voisi konfiguroida lähteviä koneita. Näin projektien konfigurointi ei olisi riippuvainen siihen kykenevistä resursseista, vaan sen voisi tehdä lähes kuka vaan vähänkään asiasta tietävä, eikä esimerkiksi loma-ajat olisi niin suuri ongelma. Lisäksi se vähentäisi inhimillisten unohdusten määrää, jota konfiguroinneissa usein ilmenee.

Jakaminen konfiguroinnin muutos vaiheisiin oli työn alkuselvittelyissä pakon sanelemaa, mutta näin jälkikäteen ajateltuna myös hyvä asia. Muutoksen pilkkominen vaiheisiin auttaa selkiyttämään kuvaa koko työkalusta ja sen tulevaisuudesta. Kun kehitys saadaan alkuun, voidaan tehdä pieniä nopeita korjauksia, jotka on helppo toteuttaa ja tuoda myönteisiä vaikutuksia parametrointiin. Myöhemmin vaikeustasoa voidaan nostaa, esimerkiksi toisistaan riippuvaisten parametrien määrittelyssä ja etsiä siihen oikea tapa havainnollistaa ja toteuttaa varsinainen toiminnallisuus. Vaikeustasoa voidaan pikkuhiljaa nostaa ja lopuksi yrittää niputtaa rakenteet sille tasolle, että niitä voidaan viedä ja tuoda konfiguraatiosta ilman ongelmia, esimerkiksi jonkun option vaikutuksesta.

Todellisen haasteen konfigurointiin heittää parametrirakenteiden hallinnointi esimerkiksi tekstissä mainitussa Excelissä. PLC-parametripolun ja parametrin arvon pitäminen yhdessä on todellinen haaste, koska Excelissä solut menevät äkkiä solmuun. Excelin hyvä puoli taas olisi sen yleinen käytettävyys ja visuaalisuus,

joka madaltaisi kynnystä sen käyttöön, etenkin parametrin arvosta vastaavan tuotehallinnan kanssa. Jos Excelissä rakenteet saisi tallennettua tietokantamaisesti, voisi myös polun ja datan sidoksen pysyminen kunnossa olla todennäköisempää.

Toinen suuri haaste liittyy konfiguraation validointiin, kun konfigurointi ohjelma tekee muutoksia taustalla. Jos ohjelma säätää joitain automatisointeja väärin, käyttäjä ei tätä välttämättä näe, joten ohjelman tekemä virhe saatetaankin huomata vasta myöhemmin, mahdollisesti asiakkaan toimesta. Validointeihin tuleekin saada mukaan kattavat ohjelmalliset automaattitestaukset konfigurointien jälkeen.

Joka tapauksessa mitä lähemmäs massaräätälöinnin ideologian (lisää sivulla 14) konfiguroinnissa päästään, sen parempi. Konfiguroitavuus itsessään on Glastonilla hoidettu hyvin, ainoastaan sen toteutus ei kaikilta osin ole pysynyt kehityksen mukana. Siihen tämä opinnäytetyö tarjoaa perusteellisen selvityksen ja vaiheittain mietityn polun, jota seuraamalla konfigurointi saadaan takaisin muun kehityksen vauhtiin.

LÄHTEET

Beckhoff Automation Oy. Scalable Industrial PC solutions. Luettu 9.9.2021. https://www.beckhoff.com/fi-fi/products/ipc/

Beckhoff Automation Oy. TwinCAT ADS. Luettu 11.9.2021. https://infosys.beckhoff.com/english.php?content=../content/1033/tcadscommon/html/tcadscommon_intro.htm&id=

Hyvönen, E. & Helokunnas, T. 2003. Ohjelmistoliiketoiminta. Helsinki: WSOY.

Juuti, T. 2008. Design Management of Products with Variability and Commonality. Contribution to the Design Science by elaborating the fit needed between Product Structure, Design Process, Design Goals, and Design Organisation for Improved R&D Efficiency. Tampere University of Technology.

Jørgensen, K.A. 2009. Product Configuration and Product Family Modelling. Department of Production, Aalborg University. Luettu: 2.8.2022. https://www.kaj.person.aau.dk/digitalAssets/199/199584_49143_productconfigurationandproductfamilymodelling.pdf

Kasanen, E., Lukka, K. & Siitonen, A. 1993. The constructive Approach in Management accounting research. Journal of management accounting research, 5(1), 243-264.

Landers, R.G., Min, B.-K. & Koren, Y. 2001. Reconfigurable Machine Tools. CIRP Annals, 50(1), 269-274.

Leon, A. 2015. Software Configuration Management Handbook. 3. painos Boston: Artech House.

Lukka, K. 2000. The key issues of applying the constructive approach to field research, in Reponen, T.(ed.), Management Expertise in the New Millennium: In Commemoration of the 50th Anniversary of Turku School of Economics and Business Administration, Series A-1.

OPC Foundation. Unified Architecture. Luettu 5.12.2021. https://opcfoundation.org/about/opc-technologies/opc-ua/

Rainamo, M. & Riikonen, M., Lasinrakentajan Käsikirja. Enterpress Oy, Tampere 1999.

Ruokavirasto. XML tekninen ohjeistus. Luettu 12.9.2021.
https://www.ruokavirasto.fi/globalassets/tietoa-meista/asiointi/oppaat-ja-lomakkeet/yhteisot/tuet-ja-kehittaminen/xml-tekninen-ohjeistus.pdf

Seth Kenlon. What is a config file? Luettu: 5.7.2022.
https://opensource.com/article/21/6/what-config-files

Suomen Tasolasiyhdistys ry. Turva- ja suojalasit. Luettu 12.8.2021.
https://www.tasolasiyhdistys.fi/lasitietoa/turva-ja-suojalasit/

Tambest. Tasomaiset Turvalasit. Luettu 12.8.2021.
https://www.tambest.fi/palvelumme/tasomaiset-turvalasit/

Tiihonen, J. & Soininen, T. 1998. Configurable products-Lessons learned from the Finnish Industry. Helsinki University of Technology.

Tseng, M.M., Jiao, J. & Merchant, M.E. 1996. Design for Mass Customization. CIRP Annals 45(1). 153–156.

Tseng, M.M., Wang, Y. & Jiao, R.J. (2019). Mass Customization. The International Academy for Production Engineering. CIRP Encyclopedia of Production Engineering. Berlin, Heidelberg: Springer. 1142–1150.

World Wide Web Consortium. Facts about W3C. Luettu 12.9.2021.
https://www.w3.org/Consortium/facts.html