

Lassi Savolainen

Siemensin ohjelmoitavien logiikoiden tiedot Red Lion -laitteille muuntavan työkalun toteutus C#- ja .NET-tekniikoilla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikka

Insinöörityö

7.5.2014

Tekijä(t) Otsikko	Lassi Savolainen Siemens ohjelmoitavien logiikoiden tiedot Red Lion -laitteille muuntavan työkalun toteutus C#- ja .NET-tekniikoilla
Sivumäärä Aika	30 sivua 7.5.2014
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Automaatiotekniikka
Suuntautumisvaihtoehto	Prosessiautomaatio
Ohjaaja(t)	Myynnin tekninen tuki Harry Nyqvist Automaatiotekniikan lehtori Antti Liljaniemi
<p>Sarlin Oy Ab tuottaa teknisiä tuotteita ja palveluja teollisuudelle sekä kunnille, ja sen osaamisalueita ovat automaatio, energia, paineilma sekä teollisuusunit. Sarlin Oy Ab on jälleenmyyjänä Red Lion -tuoteperheen laitteille. Insinööriyön tarkoituksena oli kehittää sovellus, jonka avulla pystyttäisiin helposti ja kätevästi kokoamaan Siemensin ohjelmoitaville logiikoille tehty projekti ja tuomaan projektin tiedot Red Lion -tuoteperheen laitteiden kanssa tehtyyn toteutukseen.</p> <p>Sovelluksen tarkoituksena on vähentää asiakkailta vaadittavaa käsin tehtävää konfigurointia. Ohjelman on tarkoitus myös yksinkertaistaa ja helpottaa suunnittelutyötä käytettäessä Red Lionin tuotteita yhdessä Siemensin ohjelmoitavien logiikoiden sekä Siemensin Simatic Manager- ja Tia Portal -konfigurointiohjelmien kanssa. Sovelluksen kehitysympäristönä käytettiin Microsoft Visual Studio 2010 ja ohjelmointikielenä C#. Sovelluksen käyttöliittymän suunnittelu ja toteutus tehtiin Microsoft Visual Studio 2010 käyttöliittymätyökaluja apuna käyttäen.</p> <p>Sovellusta tehtäessä perehdyttiin tarkemmin sovelluskehitykseen ja käyttöliittymän rakentamiseen. Sovelluksen ohjelmointi toteutettiin siten, että ohjelmaan olisi mahdollisimman helppo tehdä jatkokehitystyötä tarpeen ilmentyessä. Tutkimustyön perusteella käyttöliittymä suunniteltiin ja toteutettiin sovelluksen käyttötarkoitukselle mahdollisimman sopivaksi. Projektin lopputuloksena saatiin vaatimusten mukainen toimiva työkalu, joka tuottaa valmiin konfigurointitiedoston Siemensin projekteista Red Lion -tuotteille.</p>	
Avainsanat	ohjelmointi, C#, .NET, Visual Studio 2010, käyttöliittymä

Author(s) Title	Lassi Savolainen Execution of converter tool using C# and .NET techniques for converting Siemens PLC data to Red Lion devices
Number of Pages Date	30 pages 7.5.2014
Degree	Bachelor of Engineering
Degree Programme	Automation Engineering
Specialisation option	Process Automation
Instructor(s)	Harry Nyqvist, Technical sales support Antti Liljaniemi, Senior Lecturer
<p>Sarlin Oy Ab produces technical products and services for industry and municipal corporations and its line of work includes automation technology, energy, compressed air and industry ovens. Sarlin Oy Ab is a retailer for the Red Lion product family. The purpose of this thesis is to develop an application to easily and efficiently gather the information from a project made for Siemens' programmable logic controller and to bring that information for implementation done with Red Lion products.</p> <p>The purpose of the application is to reduce the amount of work needed from the customer to do the configurations for Red Lion products. The application is also meant to support marketing to help customers to transition into the Red Lion products. Microsoft Visual Studio 2010 was used as an integrated development environment for the application and the programming was done by using C#. The user interface of the application was done by using graphical user interface tools included in Microsoft Visual Studio 2010.</p> <p>Research of software development and designing graphical user interfaces was done in the beginning of the project. Programming was executed so that it is as easy as possible to continue development of the application and to add more features into the application if needed later on. The graphical user interface was designed to fulfil its purpose the best possible way. The end result of the project was a tool which filled its requirements and was used to create a ready-to-use configuration file for the Red Lion products.</p>	
Keywords	programming, C#, .NET, Visual Studio 2010, graphical user interface

Sisällys

Lyhenteet

1	Johdanto	1
2	Toimeksiantaja Sarlin Oy Ab	2
3	Simred-sovelluksen käyttöympäristö	3
3.1	Red Lion G3 series HMI	3
3.2	Red Lion G3 kadet HMI	4
3.3	Red Lion Station Plus	4
3.4	Red Lion Crimson 3 -konfigurointiohjelma	5
3.5	Siemens-ohjelmoitavat logiikat	5
3.6	Simatic Manager	6
3.7	TIA Portal	7
4	Ohjelmistokehitys	8
4.1	Asiakasnäkökulma	8
4.2	Toimittajanäkökulma	9
4.3	Vaatimukset	10
4.4	Vaatimusten toteutus	11
5	Simred-sovelluksen kehitysympäristö	11
5.1	Microsoft .NET -alusta	11
5.2	Microsoft Visual Studio 2010	12
5.3	C# ohjelmointikieli	13
6	Simred-sovelluksen suunnittelu ja toteutus	14
6.1	Vaatimukset	14
6.2	Ohjelmistoalusta ja ohjelmointikielen valinta	15
6.3	Ohjelmoinnin toteutus	16
7	Simred-sovelluksen toiminta	17
7.1	Sovelluksen käynnistäminen	17
7.2	Konfigurointitiedoston toteutus	19
7.3	Valikkorakenne	20
7.4	Konfigurointitiedoston käyttö	21

8	Käyttöliittymän suunnittelu	21
9	Simred-sovelluksen käyttöliittymän suunnittelu ja toteutus	24
9.1	Käyttöliittymän kehitys	24
9.2	Käyttöliittymän ulkoasu	27
10	Päätelmät	28
	Lähteet	29

Lyhenteet

C#	Microsoftin kehittämä ohjelmointikieli.
HMI	<i>Human Machine Interface</i> . Ihmisen ja koneen välinen vuorovaikutus.
USB	<i>Universal Serial Bus</i> . Sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen.
CF	<i>CompactFlash</i> . Muistikortti, jota käytetään kannettavissa elektroniikkalaitteissa.
OPC	<i>Open Connectivity via Open Standards</i> . Teollisuuden automaatio-sovelluksissa käytettävä tiedonsiirron standardi.
CSV	<i>Comma-Separated Values</i> . Tiedostomuoto, jolla tallennetaan taulukkomuotoista tietoa tekstitiedostoon.
SQL	<i>Structured Query Language</i> . IBM:n kehittämä standardoitu kyselykieli.
FTP	<i>File Transfer Protocol</i> . Verkkoyhteyden kautta toimiva tiedonsiirtojärjestelmä.
PLC	<i>Programmable Logic Controller</i> . Ohjelmoitava logiikka.
.NET	Windows-pohjainen ohjelmistoalusta.

1 Johdanto

Tämän insinööriyön aiheena on Sarlin Oy Ab:n toimeksi antama ohjelmointiprojekti. Ohjelmointiprojektin tarkoituksena oli kehittää sovellus, jonka avulla pystyttäisiin helposti ja kätevästi kokoamaan Siemensin ohjelmoitaville logiikoille tehty projekti ja tuomaan projektin tiedot suoraan Red Lion -tuoteperheen laitteiden kanssa tehtyyn toteutukseen. Valmiille muunnostyökalulle annettiin nimi *Simred - Siemens to Red Lion Converter Tool*.

Ennen tätä projektia Siemensin konfigurointiohjelmien sisällön tuominen Red Lionin Crimson 3 -konfigurointiohjelmaan vaati huomattavan määrän aikaa ja käsin konfiguroimista. Sovelluksen tarkoituksena on vähentää asiakkailta vaadittavaa käsin tehtävää konfigurointia ja sovelluksen on tarkoitus myös yksinkertaistaa ja helpottaa suunnittelutyötä käytettäessä Red Lion tuotteita yhdessä Siemensin ohjelmoitavien logiikoiden kanssa.

Työn suunnitteluvaiheessa perehdyttiin sovelluskehityksen ja käyttöliittymien rakennuksen teoriaan. Tämän tutkimustyön perusteella ohjelmointityötä ja käyttöliittymää tehtiin rinnatusten, jotta kokonaisuudesta tulisi mahdollisimman toimiva. Simred-sovelluksen kehitysympäristönä käytettiin Microsoft Visual Studio 2010 ja ohjelmointikielenä C#. Simred-sovelluksen käyttöliittymän suunnittelu ja toteutus tehtiin Microsoft Visual Studio 2010 käyttöliittymätyökaluja apuna käyttäen.

Sovelluksen tehtävänä on luoda konfigurointitiedosto, joka sisältää kaiken Red Lion -laitteiden tarvitseman tiedon Siemensin projektista. Konfigurointitiedosto voidaan tuoda suoraan Crimson 3 -konfigurointiohjelmalle. Crimson 3 -konfigurointiohjelma pystyy sitten lisäämään konfigurointitiedoston tiedot suoraan avoinna olevaan projektiin. Pienemmät projektit voisi yhä ohjelman valmistuttuakin viedä käsin Crimson 3 -konfigurointiohjelmaan, mutta suuremmat projektit ovat niin paljon työläämpiä siirtää, että niissä sovelluksen tarpeellisuus todella korostuu.

Työn kirjallisessa osuudessa esitellään aluksi insinööriyön tuloksena syntyneen sovelluksen taustaa ja sen käyttöympäristöä. Lisäksi käsitellään ohjelmistokehitystä kokonaisuutena ja tarkemmin Simred-sovelluksen kehitysympäristöä, suunnittelua ja toteu-

tusta. Työssä käsitellään myös sovellusten käyttöliittymän suunnittelua yleisesti sekä Simred-sovelluksen näkökulmasta.

2 Toimeksiantaja Sarlin Oy Ab

Sarlin on kotimainen perheyhtiö, joka on perustettu vuonna 1932. Sarlin tuottaa teknisiä tuotteita ja palveluja teollisuudelle sekä kunnille, ja sen osaamisalueita ovat automaatio, energia, paineilma sekä teollisuusunit. Sarlin koostuu kolmesta eri liiketoimintayksiköstä, jotka ovat Sarlin Oy Ab, Sarlin Furnaces AB ja Beamex Oy Ab, yhdessä ne työllistävät 260 henkilöä. Omavaraisuusaste Sarlinilla vuonna 2013 oli lähes 50 % ja liikevaihto yli 60 miljoonaa euroa. Sarlin toteuttaa asiakkailleen sopivimmat teknologiaratkaisut ja hoitaa niiden ylläpidon ja huollon. Sarlin palvelee useassa eri myynnin, huollon sekä automaation jälleenmyynnin toimipisteessä ympäri Suomen. [1.]

Sarlinin automaation osaamisalue koostuu koneautomaatiosta, prosessiautomaatiosta, koneturvallisuudesta, tietoliikenteestä sekä ohjaus- ja näyttötekniikasta. Paineilman osaamisalueella Sarlinilla on markkinoiden laajin tuote- ja palveluvalikoima, johon kuuluu muun muassa kompressorit, paineilman jälkikäsittely, Sarlin Balance - paineilmajärjestelmän hallinta, erikoissovellukset, huolto ja korjaukset, Sarlin Syyni - paineilmajärjestelmän mittaus ja analysointi, Sarlin Optimi - paineilmajärjestelmän hallintapalvelu sekä Sarlin Tuotto paineilman tuottopalvelu. Sarlinin energian osaamisalue tarjoaa ympäristöystävällisiä energiaratkaisuja, joita ovat mikroturbiinilaitokset, moottorivoimalaitokset, biokaasulaitokset, maa- ja biokaasun tankkausasemat sekä biokaasun jalostuslaitokset. Sarlin Furnaces valmistaa, kehittää, modernisoi ja huoltaa teollisuusuneja, joita käytetään metallien lämpökäsittelyyn. Sarlin tarjoaa myös kaasuanalysaattoreita sekä ratkaisuja nesteiden suodattamiseen ja desinfointiin. [1.]

3 Simred-sovelluksen käyttöympäristö

Simred-sovellus on suunniteltu käytettäväksi Red Lion -tuoteperheen operointipaneelien ja Data Station Plus -aseman sekä Siemensin ohjelmoitavien logiikoiden kanssa. Red Lion -tuoteperheen laitteita käytetään Crimson 3 -konfigurointiohjelman avulla. Siemensin ohjelmoitavien logiikoiden kanssa käytetään Simatic Manager tai Tia Portal ohjelmistoja.

3.1 Red Lion G3 series HMI

Red Lion G3 -sarjan operointipaneelit sisältävät monipuoliset ominaisuudet eri automaatiototeutuksiin. Kaikki G3-sarjan operointipaneelit sisältävät näytön lisäksi Ethernet- ja USB-portit, CF-korttipaikan ja tuen viidelle sarjaporttiliitännälle, joihin pystyy liittämään esimerkiksi ohjelmoitavia logiikoita, lämpötilasäätimiä tai muita sarjaportteihin kytkettäviä laitteita. G3 -sarjan operaatiopaneeleita on monia eri malleja, mutta eri mallit eroavat toisistaan lähinnä fyysisen kokonsa ja näytön koon sekä näytön ominaisuuksien sekä ohjelmoitavien näppäimien lukumäärän puolesta. Paneelia käytetään mallista riippuen joko ohjelmoitavien näppäimien avulla tai kosketusnäytön ja ohjelmoitavien näppäimien yhdistelmällä. [2; 3; 4.]

Red Lion G3 -sarjan operointipaneelien kommunikointi-ominaisuuksiin kuuluu yli kahden sadan protokollan muunto Ethernet- ja sarjaporttien välillä sekä mahdollisuus yhdeksän eri protokollan samanaikaiseen ajoon. Tiedonsiirto paneeleilta ylempiin järjestelmiin toteutetaan OPC-rajapinnalla. Yhdellä OPC-rajapinnalla kyetään liittämään koko laitekanta ylempiin järjestelmiin. [2.]

Red Lion G3 -sarjan operointipaneeleilla pystytään tallentamaan tietoa paneeliin kytketyistä muuttujista. Tiedonkeräys voidaan toteuttaa suoraan CF-muistikortille CSV-tiedostomuotoon tai vaihtoehtoisesti suoraan SQL-tietokantaan ilman erillistä ohjelmistoa. [2; 3; 4.]

Operointipaneeleissa on sisäänrakennettu verkkopalvelin, joka mahdollistaa operointipaneelin käytön tavallisella verkkoselaimella. Järjestelmään kirjaudutaan sisään käyttäjätunnuksilla ja salasanalla. Verkkopalvelimeen on myös mahdollista saada yhteysrajoi-

tukset määrätystä IP-osoitteesta tai verkosta. Paneeli kykenee kaksisuuntaiseen yhteyteen tekstiviestien tai sähköpostien avulla. Paneeli kykenee lähettämään hälytysviestejä tai laajennettuja raportteja ja myös laitteelle kirjoittaminen on mahdollista. Operaatiopaneelilta pystytään myös ajastettuun tiedon siirtoon FTP-yhteyden avulla, jolloin tiedostot siirretään automaattisesti paneelilta FTP-palvelimelle ajastuksen mukaisesti. [2; 3.]

G3-sarjan operaatiopaneelien projektit luodaan Crimson 3 -konfigurointiohjelmalla. Projektit toteutetaan graafisella käyttöliittymällä, josta löytyy oma symbolikirjasto ja monikielisyystuki kahdellekymmenelle kielelle samaan projektiin. [2.]

3.2 Red Lion G3 kadet HMI

Red Lion G3 kadet -sarjan operaatiopaneelit ovat hieman karsitumpia versioita G3-sarjan operaatiopaneeleista. G3 kadet -paneelissa ei ole ollenkaan ohjelmitavia näppäimiä, sillä niitä käytetään ainoastaan kosketusnäytön avulla. Ominaisuuksiensa puolesta G3 kadet -sarjan operointipaneelit ovat hyvin samankaltaisia G3-sarjan paneelien kanssa, mutta niissä ei ole USB-portteja, tiedonkeräystä eikä verkkopalvelin ominaisuutta ja niitä ei voi laajentaa laajennuskorteilla samalla tavalla kuin G3-sarjan paneeleita. [4; 5.]

3.3 Red Lion Station Plus

Red Lion Data Station Plus -asemat eivät sisällä ollenkaan operointipaneelia, sillä ne keskittyvät protokollan muunto- ja kommunikointi-ominaisuuksien tarjoamiseen tilanteissa, joissa paneeliominaisuudet eivät ole tarpeellisia. Kaikki Data Station Plus -asemat sisältävät samanlaiset liitännät, laajennusmahdollisuudet ja protokollan muunto-ominaisuudet G3-sarjan operointipaneelien kanssa. Data Station Plus SX- ja Data Station Plus GT- mallit sisältävät lisäksi vastaavat tiedonkeräys, verkkopalvelin ja etäyhteysominaisuudet kuin G3 -sarjan operointipaneelit. Data Station Plus GT sisältää lisäksi laajennetun muistin laajempienkin ohjelmien rakentamiseen. [2; 6; 7.]

3.4 Red Lion Crimson 3 -konfigurointiohjelma

Red Lion Crimson 3 -konfigurointiohjelma on lisenssivapaa Windows-ohjelma sovellusten luontiin Red Lion -laitteille. Ohjelma mahdollistaa oman C-ohjelmointikieleen perustuvan ohjelman rakentamisen sekä valmiit funktiot kommunikointiin, tiedonkeräykseen ja tiedostojen käsittelyyn. Crimon3-konfigurointiohjelman graafisella käyttöliittymällä pystyy myös luomaan paneeleille erilaisia näyttöjä, joiden tekemiseen voi käyttää omia kuvatiedostoja tai valmista symbolikirjastoa, joka sisältää 4000 eri symbolia. Ominaisuuksista löytyy myös mahdollisuus luoda paneelille animaatioita. Kuvassa 1 näkyy Crimson 3 -konfigurointiohjelman käyttöliittymä ja paneelin näyttöjen luomiseen tarkoitettuja työkaluja ja objekteja. [2.]



Kuva 1. Crimson 3 -konfigurointiohjelman käyttöliittymä.

3.5 Siemens-ohjelmoitavat logiikat

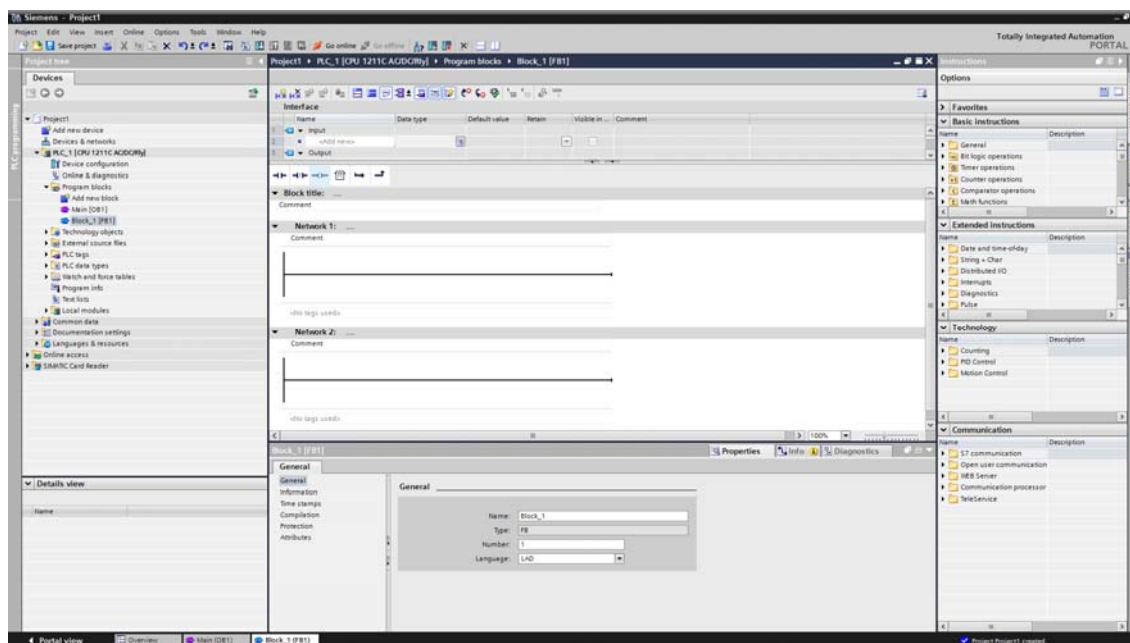
Ohjelmoitavat logiikat kulkevat usein nimellä PLC eli Programmable Logic Controller. Ohjelmoitava logiikka on mikroprosessoripohjainen laite, jota käytetään reaaliaikaisten automaatioprosessien ohjauksissa. Ohjelmoitavia logiikoita käytetään esimerkiksi tehtaiden ohjauksien ja säätöjen toteuttamiseen tai vaikka yksittäisten koneiden ohjaamiseen. [8.]

Ohjelmoitavat logiikat ovat nykyajan automaation perusta. Ne ovat kehittyneet releiden ja ajastimien korvaajista monipuolisiksi ohjauskeskuksiksi. Siemensiltä löytyy ohjelmoitavien logiikoiden eri malleja, joiden sopivuus eri toteutuksiin riippuu toteutuksen laajuudesta sekä siitä millaiseen tilaan ja käyttöön logiikkaa ollaan hankkimassa. Siemens ohjelmoitavat logiikat koostuvat eri moduuleista, jotka lopulta määrittävät mihin toimintoihin ohjelmoitava logiikka pystyy. [8.]

Ohjelmoitavat logiikat tarvitsevat toimiakseen logiikkaohjelman, jonka perusteella logiikka toteuttaa prosessin säädön ja tarvittavan ohjauksen. Siemensin logiikoille ohjelmat tehdään Simatic Manager- tai Tia Portal -työkalujen avulla. [8.]

3.6 Simatic Manager

Simatic Manager on Siemensin kehittämä konfigurointi- ja ohjelmointityökalu. Simatic Manageriksi kutsutaan graafista käyttöliittymää, jolla käytetään SIMATIC STEP 7 -työkalua. Simatic Managerin avulla pystytään tekemään Siemensin ohjelmoitaville logiikoille kaikki tarpeellinen suunnittelu, konfigurointi, monitorointi ja ohjelmointityö. Siemens logiikkamallien tavoin myös Simatic Manager on modulaarinen. Simatic Manager käytännössä sitoo yhteen kaikki tarvittavat työkalut, joita prosessinohjausjärjestelmässä tarvitaan ja tarvittaessa siihen pystytään asentamaan lisää työkaluja, jotta se kykenee kaikkeen tarvittavaan toimintaan. Simatic Managerista on tullut jo korvaava versio uudemmille ohjelmoitaville logiikoille nimeltä TIA Portal. Pääosin ennen vuotta 2007 valmistuneiden ohjelmoitavien logiikoiden kanssa käytetään Simatic Manageria ja sitä uudempien ohjelmoitavien logiikoiden kanssa TIA Portal -työkalua. Kuvassa 2 Simatic Managerin käyttöliittymä. Käyttöliittymän takimmaisessa ikkunassa näkyy projektin tiedostopuun rakenne, ja edessä olevassa ikkunassa näkyy avattuna symbolieditori. [9; 10; 11; 12.]



Kuva 3. TIA Portal -käyttöliittymä.

4 Ohjelmistokehitys

Ohjelmistojen tuottaminen organisoidaan tyypillisesti projekteiksi. Ohjelmistokehitysprojekteissa on yleensä kaksi osapuolta, asiakas ja toimittaja. Asiakas on ohjelmiston tilaaja ja toimittaja on ohjelmiston kehittäjä. Ohjelmistokehitysprojekteissa toteutetaan asiakkaalta saatujen vaatimusten perusteella haluttu ohjelmisto. Ohjelmistokehitysprojektiä on hyvä tarkastella sekä asiakasnäkökulmasta että toimittajanäkökulmasta. [16, s.19.]

4.1 Asiakasnäkökulma

Asiakkaalla on yleensä selkeät liiketoimintatavoitteet ja siksi asiakas näkee projektin toimittajaa laajempuna kokonaisuutena. Asiakkaan tavoite on kehittää yrityksen toimintaa liiketoimintaympäristön kannalta paremmaksi. Toimintaa kehitetään eri kehittämissankkeilla ja projekteilla. Asiakas voi jakaa hankkeet pienempiin osaprojekteihin, joita voivat olla esimerkiksi esitutkimus, määrittely ja käyttöönotto. [16, s.19 - 20.]

- Esitutkimuksessa asiakas kartoittaa vaihtoehdot, tekee alustavan määrittelyn, tarveanalyysin, kannattavuuden ja riskien arvioinnin sekä päätöksen seuraavan vaiheen käynnistämisestä.
- Määrittelyprojektissa määritellään lopputuloksen toiminnallisen määrittelyn, jonka avulla mahdollistetaan työmääräarvioiden tekemisen, riskien arvioinnin ja päätöksen seuraavan vaiheen käynnistämisestä.
- Toteutusprojektissa tehdään toteutuksen tekninen määrittely, yksityiskohmainen suunnittelu, ohjelmoinnin toteutus ja merkittävän osan testausta.

Asiakas päättää mitä tekee itse ja mikä tilataan ulkopuolelta. Yllämainittujen osaprojektien lisäksi asiakkaalla voi olla käynnissä muita osaprojekteja, kuten esimerkiksi mahdollisten toimittajien kilpailutus tai ylläpito ja käyttötuki. Yhteistä projekteille on kuitenkin, että ne määrittävät sisällön ja tavoitteet. Tavoitteiden saavuttamiseksi projektille annetaan käyttöön joukko resursseja, joiden avulla projektin läpivieminen on mahdollista. Resurssit voivat olla esimerkiksi työhön osallistuvia henkilöitä tai rahaa alihankintaan. Lisäksi projektilla on oltava aikataulu. [16, s. 20 - 21.]

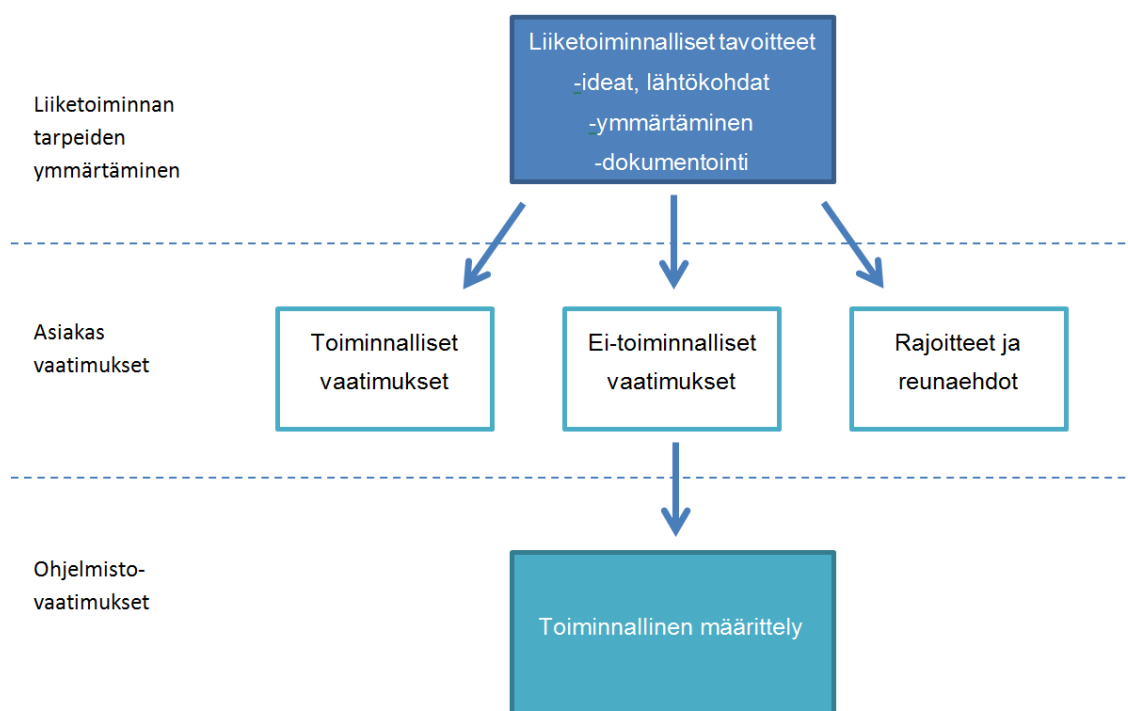
4.2 Toimittajanäkökulma

Toimittajalle ohjelmistoprojektin lähtökohta on asiakkaalta tulevat vaatimukset, jotka kuvaavat projektin sisällön mahdollisimman tarkasti asiakkaan kannalta. Täydelliset vaatimusmäärittelyt asiakkaan puolelta ovat lähes mahdoton toteuttaa, joten vaatimukseen tulee lähes aina lisäyksiä, tarkennuksia ja muutoksia koko projektin ajan. Näitä muutoksia aiheuttavat usein yrityksen omiin lähtökohtiin ajan myötä tulevat muutokset. Myös monet toteutuksen yksityiskohdat voivat aiheuttaa muutoksia. Yleensä tyydytään ”riittävän tarkkoihin” vaatimukseen, joihin poikkeuksen kuitenkin tuovat turvallisuuskriittisimmät järjestelmät, kuten lentokoneet tai ydinvoimalat, joissa kaikki muutosten aiheuttamat riskit halutaan minimoida. [16, s. 21 - 22.]

Toimittajalle ongelmallisimpiin asioihin kuuluu yleensä tasapainoilu asiakaskohtaisuuden ja tuotokeskeisyyden välillä. Mitä tarkemmin asiakkaan vaatimukset otetaan huomioon, sitä tyytyväisempi asiakas yleensä on. Tämä kuitenkin usein lisää ylläpidon tarvetta ja aiheuttaa mahdollisesti lisätyötä ohjelmoidessa, joka aiheuttaa toimittajalle lisäkustannuksia. [16, s. 21 - 22.]

4.3 Vaatimukset

Vaatimuksilla tarkoitetaan asioita, joita tuotteella pystyy tekemään ja ominaisuuksia, joita tuotteella tulee olla. Vaatimukset voidaan tavallisesti luokitella kolmeen luokkaan: Toiminnallinen vaatimus, ei-toiminnallinen vaatimus ja reunaehdot. Toiminnallisilla vaatimuksilla tarkoitetaan ohjelman toimintaan liittyviä asioita, esimerkiksi järjestelmän kielen vaihtaminen yhdellä painikkeella. Ei-toiminnalliset vaatimukset tarkoittavat ei-toiminnallisia asioita, joita ohjelman täytyy sisältää, esimerkiksi ohjelman käyttöohje. Reunaehdoilla tarkoitetaan tiettyjä vaatimuksia joiden pitää täytyä, esimerkiksi ohjelman täytyy olla toteutettu Windows-ympäristöön C#-ohjelmointikielellä. Kuvassa 4 näkyy kaavio asiakas- ja ohjelmistovaatimuksista. [16, s. 61.]



Kuva 4. Asiakas- ja ohjelmistovaatimukset [16, s. 62].

4.4 Vaatimusten toteutus

Asiakasvaatimukset toteutetaan ohjelmistovaatimuksilla. Ohjelmistovaatimukset ovat ohjelman toimintoja, jotka määrittävät kuinka asiat esitetään käyttäjälle ohjelmassa. Ohjelmistovaatimukset kuvautuvat toteutustasolla joukoksi teknisiä vaatimuksia. [16, s. 62 - 63.]

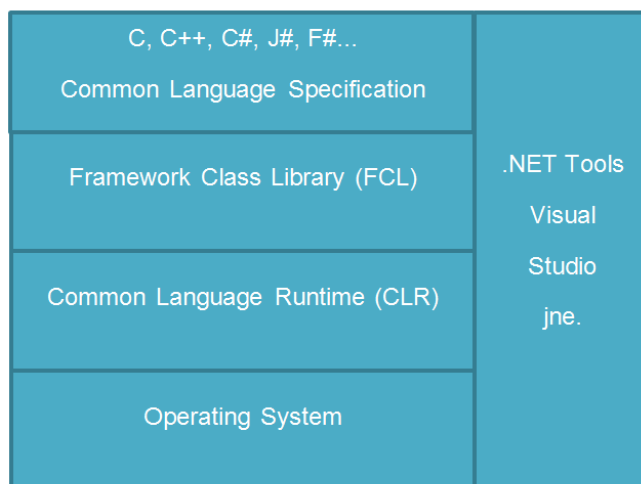
5 Simred-sovelluksen kehitysympäristö

5.1 Microsoft .NET -alusta

Microsoft .NET -alusta on Microsoftin kehittämä ohjelmistoalusta, joka tarjoaa työkalut ja teknologiat erityyppisten sovellusten tekemiseen PC-järjestelmissä ja mobiililaitteissa. Microsoft .NET -arkkitehtuuri koostuu seuraavista osista: Common Language Specification, Framework Class Library (FCL), Common Language Runtime (CLR) sekä palvelualustasta ja .NET-ohjelmistotyökaluista. [17, s. 3.]

Common Language Specification on alusta, joka yhdistää eri ohjelmointikielien, että kaikilla kielillä on samat kielispesifikaatiot. Näin pystytään käyttämään eri ohjelmointikieliä samassa ympäristössä. Framework Class Library on laaja kokoelma luokkia ja erilaisia tietorakenteita. Luokkien avulla .NET-sovelluksessa saadaan toteutettua eri toimintoja kuten esimerkiksi tiedostojen käsittelyä, tietokantayhteyden graafisten käyttöliittymien toteutus ja grafiikan piirto. Common Language Runtime on ajoympäristö, joka kääntää ja tulkitsee välitason kielelle (Intermediate Language, IL) käännetyn ohjelmakoodin binäärimuotoon, jota tietokoneiden prosessorit ymmärtävät. CLR toimii .NET sovellusohjelman ja käyttöjärjestelmän välissä käyttöliittymänä. Microsoftin .NET-työkalut ovat erilaisia ohjelmiston kehitystyökaluja, joita on kehitetty .NET-alustan ohjelmistokehitystä varten. .NET-ohjelmistotyökalut (esimerkiksi Microsoftin Visual Studio) on kehitetty .NET-alustalla tapahtuvaa ohjelmistonkehitystä varten. [17, s. 4 - 5; 18, s. 13.]

Perinteisten Windows-ohjelmien eli niin kutsuttujen Stand-alone -tyyppisten sovellusten toteuttamiseen .NET-alustan käyttäjämallina käytetään .NET-sovelluskehystä (.NET Framework) ja Windows Forms -nimistä graafista ohjelmointirajapintaa. Kuvassa 5 näkyy Microsoftin .NET-ohjelmistoarkkitehtuuri. [17, s.5; 18, s.15.]

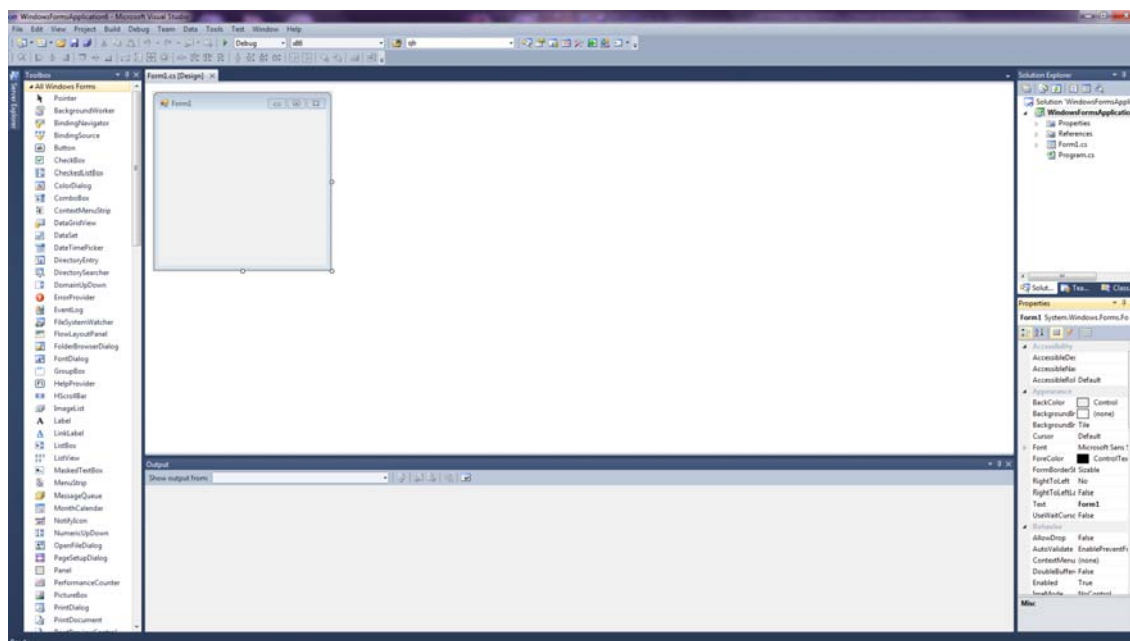


Kuva 5. Microsoft .NET -ohjelmistoarkkitehtuuri [17, s. 3].

5.2 Microsoft Visual Studio 2010

Microsoft Visual Studio on Microsoftin kehittämän .NET-alustan ohjelmistonkehitys työkalu. Sen tarkoitus on nopeuttaa uusien ohjelmistojen valmistumista tarjoamalla valmiiksi rakennettuja osia ohjelmoinnin tueksi. Tämä nopeuttaa ja helpottaa ohjelman kehitystyötä koska kaikkea ei tarvitse rakentaa alusta asti. [18, s.12.]

Visual Studio tarjoaa mahdollisuuden käyttää useita eri ohjelmointikieliä kuten C#, J#, C++, Visual Basic. Ohjelmoinnin lisäksi Visual Studiota voi käyttää myös graafisten käyttöliittymien luomiseen. Lisäksi Visual Studiolla pystyy esimerkiksi kääntämään ohjelman lähdekoodin konekielelle sekä julkaisemaan valmiita ohjelmia. Visual Studio 2010 tukee suurimmillaan .NET-sovelluskehityksen versiota 4.0, mutta tuki myös vanhempien versioiden käytölle löytyy. Kuvassa 6 näkyy Visual Studio 2010 -käyttöliittymä ja oman sovelluksen käyttöliittymän rakennukseen käytettäviä työkaluja. Työkaluikkuna näkyy kuvassa vasemmalla, käyttöliittymä rakennetaan kuvan keskellä näkyvään tyhjäan ikkunaan.



Kuva 6. Microsoft Visual Studio 2010 -käyttöliittymä.

5.3 C# ohjelmointikieli

C# (CSharp) on Microsoftin vuonna 2000 julkaisema ohjelmointikieli. Se kehitettiin yhdistämään C++-ohjelmointikielen tehokkuus ja Java-ohjelmointikielen helppokäyttöisyys. C#-ohjelmointikielen ominaisuudet edistävät varmempien ja toimivampien ohjelmien luomista. C#-ohjelmointikieli on ensimmäinen C-kieliperheen komponenttisuunniteltu ohjelmointikieli ja se on niin sanottu puhdas olio-kieli. C# toimii Microsoftin .NET-alustan pääohjelmointikielenä. [17, s.5; 19, s. 1.]

6 Simred-sovelluksen suunnittelu ja toteutus

6.1 Vaatimukset

Työ aloitettiin tekemällä tarvekartoitus, jonka perusteella tehtiin vaatimusmäärittely. Vaatimusmäärittelyssä määriteltiin sovelluksen sisältö, toiminnot, käyttöliittymän vaatimukset ja sovelluksen luoman konfigurointitiedoston sisältö sekä reunaehdot.

Simred-sovellus haluttiin kehitettävän Windows-käyttöjärjestelmälle, mutta muuten sovelluksen ohjelmointikielen ja kehitysympäristön puolesta annettiin vapaat kädet. Simred-sovelluksen haluttiin toimivan Siemensin Simatic Manager- ja TIA Portal -ohjelmien kanssa. Molempien ohjelmien haluttiin toimivan käyttäjän näkökulmasta samalla tavalla. Siemensin ohjelmien tiedot haluttiin siirrettäväksi kahdella tavalla. Tietojen tuli olla suoraa kopioitavissa Siemensin ohjelmien projekteista tai tuotavissa Simred-sovellukseen CSV-tiedostona. Ohjelmasta piti pystyä valitsemaan mistä sijainnista muunnettava tiedosto avataan.

Simred-sovellusta haluttiin käytettävän yksinkertaisella ja selkeällä käyttöliittymällä. Käyttöliittymän oletussivulla haluttiin olevan kaikki tärkeimmät kontrollit näkyvillä ja helposti käytettävissä. Näytöllä olevien kontrollien määrä haluttiin silti pitää minimissään, jotta näyttö pysyisi selkeänä. Siemensin konfigurointiohjelmita lisättävien tietojen haluttiin olevan koko ajan näkyvillä käyttöliittymässä muuntamiseen asti. Muunnoksen valmistuttua valmiin konfigurointitiedoston tiedot haluttiin näkyvän käyttöliittymässä seuraavaan muunnokseen tai ohjelman sammutukseen asti.

Käyttöliittymään haluttiin kenttä johon pystyy antamaan nimi ohjelmoitavalle logiikalle Crimson 3 -projektia varten tai jos ei nimeä anneta, antaa sovellus ohjelmoitavalle logiikalle vakionimen. Siemensin tiedoista haluttiin tuoda erityisesti symbolitaulut ja data blockit helposti Simred-sovellukseen. Data blockeille tuli pystyä syöttämään data blockin numero sekä määrittämään asetetaanko data blockit omiin kansioihin vai symbolien kanssa samaan kansioon Crimson 3 -konfigurointiohjelmassa. Symbolitauluista ja data blockeista piti pystyä tuomaan vain halutut muuttujat ja siirron tuli silti toimia oikein. Sovelluksen tuli laskea data blockien osoitteet oikein Crimson 3 -konfigurointiohjelmalle huolimatta siitä kuinka monta data blockia tuodaan ja mitä osia data blockista ollaan tuomassa. Symbolitaulujen ja data blockien muuttujille tarvittiin eri

toimintaohjeet jokaiselle eri muuttuja tyypille erikseen. Muuttaja tyypit ja niille tarvittavat toiminnot listattiin erikseen.

Simred-sovelluksen haluttiin muokkaavan kaikki Siemens -ohjelmien tiedot Red Lion -laitteille soveltumaan muotoon. Simred-sovelluksen tuli "siivota" Crimson 3 -konfigurointitiedosto kaikkien epäsopivien merkkien osalta. Lisäksi Simred-sovelluksen tuli täyttää kaikki Crimson 3 -konfigurointitiedot oletustiedoiksi, niiltä osin joille käyttäjä ei ole valinnut asetuksia. Konfigurointitiedostolle tehtiin erillinen listaus tiedoista, joita sen täytyi sisältää sekä tarkat määrittelyt missä muodossa eri tietojen tarvitsi olla.

Ohjelmoinnin puolelta vaatimuksena oli ohjelman tarkka kommentointi ja selkeä sekä suoraviivainen ohjelmointi.

6.2 Ohjelmistoalusta ja ohjelmointikielen valinta

Nykyohjelmistotuotannon suuntaus on, että uusia ohjelmistoja ei määritellä, suunnitella ja toteuteta kokonaan puhtaalta pöydältä, vaan ne perustuvat johonkin ohjelmistoalustaan. Toteutusta suunnitellessa oli varsin helppo päätös valita valmis ohjelmistoalusta myös tätä toteutusta varten. On huomattavasti helpompi hyödyntää valmiita työkaluja kuin aloittaa täysin puhtaalta pöydältä. [16, s. 23.]

Lähtökohtana oli Windows-käyttöjärjestelmällä toimiva sovellus, mutta muuten ohjelmistoalustan ja ohjelmointikielen valinta oli vapaa. Käyttöjärjestelmän ollessa Windows, tuntui luontevalta valita ohjelmistoalustaksi Visual Studio. Myös oma kokemus Visual Studion käytöstä ja jo valmiiksi käytössä oleva Visual Studio 2010 helpottivat päätöstä.

Visual Studio tukee monia ohjelmointikieliä, mutta tuetuista ohjelmointikielistä eniten omaa kokemusta löytyy C#-ohjelmointikielestä. Lisäksi C#-ohjelmointikielen helppokäyttöisyys ja tehokkuus yhdistettynä monipuoliseen kirjallisuuteen sekä verkosta löytyviin esimerkkeihin ja malleihin, auttoivat valitsemaan sovelluksen ohjelmointikieleksi C#.

6.3 Ohjelmoinnin toteutus

Ohjelmointityö aloitettiin tutustumalla tarkasti aiheeseen ja pohtimalla erilaisia ohjelmoinnin toteutustapoja ja malleja, joita voisi projektissa hyödyntää. Ennen ohjelmoinnin aloittamista piti myös tarkkaan määrittellä ohjelmointityyli, muuttujien nimeäminen ja systemaattinen kommentointitapa. Hyvällä määrittelyllä ja huolellisella suunnittelulla haluttiin varmistaa laadukas ja toimiva ohjelma, ja samalla haluttiin säästyä monelta myöhemmältä ongelmalta ja vaikeudelta. Myös Sarlinilta toivottiin selkeää ja suoraviivaista ohjelmointia, jotta ohjelman helppo jatkokehitys olisi mahdollista. Jatkokehitys voisi tulla kysymykseen jos asiakkailta tulee toivomuksia uusista ominaisuuksista, tai sovelluksen käyttötarkoitus muista syistä laajenee tai muuttuu.

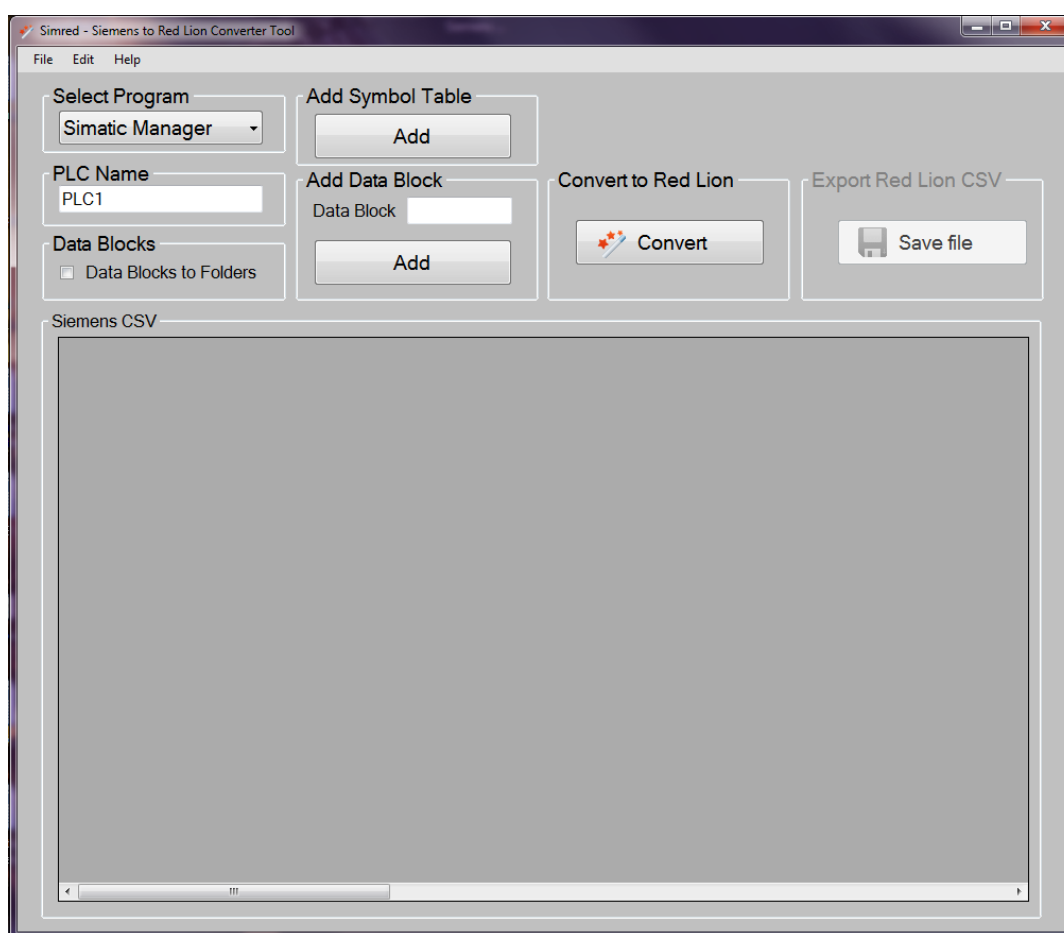
Ohjelmistojen tuotantoa varten on olemassa lähes rajaton määrä erilaisia lähestymistapoja ja valmiita ohjelmistoprojektimallejakin on valtavasti. Tavallisen ohjelmointiprojektin tapaan jo projektin alkuvaiheessa vaatimusmäärittelyyn tuli lisäyksiä ja tarkennuksia. Tämän vuoksi sekä projektiin luonteesta johtuen, päädyttiin käyttämään yksinkertaisinta mallia, jota kutsutaan nimellä "code-and-hack". "Code-and-hack" tarkoittaa lähestymistapaa, jossa ohjelmaa yksinkertaisesti kasvatetaan, laajennetaan, muutetaan ja korjataan kunnes se tyydyttää asiakasta. Sovelluksen pohjana käytettiin alkupeleistä vaatimusmäärittelyä, mutta lisäykset ja muutokset toteutettiin tarpeen ilmentyessä. [16, s. 29.]

Ohjelmointityötä tehtäessä hyödynnettiin aiemmin kerättyä ohjelmointikokemusta varsin laajasti. Ohjelmointityössä hyödynnettiin monipuolisesti myös Visual Studio-ohjelmistonkehitys työkalun valmiita kirjastoja. Vaikeimpien toteutuksien ja ongelmatilanteiden ratkaisuun hyödynnettiin kirjallisuutta sekä verkosta löytyviä malleja ja esimerkkejä.

7 Simred-sovelluksen toiminta

7.1 Sovelluksen käynnistäminen

Simred-sovellus toimii käyttäjän näkökulmasta samalla tavalla, riippumatta siitä kumpi Siemensin ohjelma hänellä on käytössään. Käyttäjä pystyy myös tekemään asetusten valinnat vapaavalintaisessa järjestyksessä. Tärkeimmät kontrollit löytyvät oletussivun käyttöjärjestelmän keskellä olevan taulukon yläpuolelta. Kaikkiin ohjelman asetuksiin ja toimintoihin pääsee kuitenkin käsiksi myös käyttöliittymän yläreunassa olevista alasve-tovalikoista. Kuvassa 7 näkyy Simred-sovelluksen käyttöliittymän oletusnäköymä.



Kuva 7. Simred-sovelluksen käyttöliittymän oletusnäköymä.

Käyttäjän käynnistettyä Simred-sovellus aukeaa hänelle oletusnäköymä Simred-käyttöliittymästä. Käyttäjä valitsee Siemensin ohjelmatyyppin, joka hänellä on käytös-sään oletusnäköymässä olevasta valintaikkunasta. Crimson 3 -konfigurointiohjelmassa

ohjelmoitavien logiikoiden täytyy olla nimettyjä. Oletuksena Simred-sovellus antaa ohjelmoitavalle logiikalle nimen PLC1. Mikäli käyttäjä haluaa nimetä ohjelmoitavan logiikan eri tavalla, voi hän vaihtaa PLC nimi kenttään haluamansa nimen.

Tietojen tuomiseen Siemensin ohjelmista käyttäjällä on kaksi eri vaihtoehtoa sekä näiden vaihtoehtojen yhdistelmä. Käyttäjä voi tuoda Siemens projektin tiedot erillisessä CSV-tiedostossa, jolloin yläreunan alavetovalikon import-näppäimellä pääsee valitsemaan tietokoneen tiedostoista haluamansa CSV-tiedoston. Valitun tiedoston tiedot latautuvat Simred-sovelluksen oletusnäkyvän taulukkoon heti käyttäjän tiedostovalinnan jälkeen. Toisena vaihtoehtona käyttäjä voi kopioida tiedot suoraan Siemensin Simatic Manager- tai Tia Portal -käyttöliittymistä Simred-sovelluksen taulukkoon. Mikäli tiedot tuodaan suoraan kopioimalla, lisätään symbolitaulut lisää symbolitaulu -näppäimellä ja data blockit lisää data block -näppäimellä. Data blockeja lisätessä täytyy määrittää myös data blockin numero, joka määritellään lisää data block -valikossa olevaan kenttään.

Simred-sovellus tunnistaa mitä Siemens ohjelmien tietoja missäkin kohtaa taulukkoa on automaattisesti solujen sisällöstä. Erityyppiset tiedot erotetaan taulukossa tyhjästä rivistä tietojen välissä. Näin tietoja pystyy lisäämään taulukkoon haluamassaan järjestyksessä. Jos jostain välistä sattuu unohtumaan jotain tietoja pois voi ne lisätä vaikka taulukon loppuun ja ohjelma silti tunnistaa mistä tiedoista on kyse ja laittaa ne oikeaan paikkaa Crimson 3 -konfigurointitiedostoon.

Simred-sovelluksen taulukossa näkyviä tietoja pystyy vapaasti selaamaan taulukon omilla vierityspalkeilla tai vaihtoehtoisesti näppäimistön nuolinäppäimillä tai hiiren rullaa apuna käyttäen. Taulukkoon voi halutessaan lisätä rivejä hiiren oikean näppäimen alta löytyvän valikon kautta tai vaihtoehtoisesti näppäimistön pikakomentojen avulla. Myös taulukon yksittäisten solujen sisältöjä voi muuttaa, mikäli käyttäjä haluaa vielä tehdä muutoksia muunnettaviin tietoihin.

Data blockien muunnosta varten täytyy Simred-sovelluksen laskea data blockeille niiden oikeat osoitteet. Siemensiltä siirrettävien tietojen joukosta löytyy data blockeille sarake, jossa määritellään jokaisen data blockin muuttujan tyyppi. Tämän tyyppin avulla lasketaan jokaiselle muuttujalle oma osoitteensa Crimson 3 -konfigurointiohjelmaa varten. Crimson 3 -konfigurointiohjelmassa jokaisen yksittäisen data blockin sisältö voi olla

omissa kansioissaan, tai sitten yhdessä muiden muuttujien kanssa. Simred-sovelluksen data blocks to folders -valinnalla päätetään kumpaan muotoon data blockit muunnetaan Crimson 3 -konfigurointiohjelmalle.

Käyttäjän koottua siirrettävät tiedot ja valittua haluamansa muutokset Simred-sovelluksen käyttöliittymästä, voi käyttäjä aloittaa tietojen muuntamisen Crimson 3 -konfigurointiohjelmalle sopivaan muotoon. Muunto aloitetaan painamalla convert-näppäintä.

7.2 Konfigurointitiedoston toteutus

Crimson 3 -konfigurointiohjelma ja Red Lionin laitteet, joita varten konfigurointitiedosto tehdään, käyttävät eri merkistöä kuin Siemensin ohjelmistot ja ohjelmoitavat logiikat. Lisäksi Red Lionin tuotteilla on myös muita rajoituksia muun muassa muuttujien nimeämisissä. Käyttäjän aloittaessa tietojen siirtämistä Red Lionin tuotteille sopivaan muotoon, tarkistaa Simred-sovellus muunnettavat tiedot. Simred-sovellus tekee tarvittavat korjaukset Siemensiltä siirrettäviin tietoihin automaattisesti. Jos muunnettavien tietojen joukossa on vielä sellaisia virheitä, joihin käyttäjän täytyy puuttua tai muita asioita, joihin käyttäjän vielä tarvitsee tehdä valintoja, ohjaa Simred-sovellus käyttäjän tekemään tarvittavat lisäykset.

Simred-sovelluksen saatua muunnos valmiiksi tulee uusi Crimson 3 -konfigurointitiedosto näkyviin oletusnäkyvän taulukkoon. Taulukkoa pääsee vielä tässäkin vaiheessa vapaasti tarkastelemaan samalla tavoin kuin ennen muunnosta sekä korjaamaan solujen tietoja. Suurin osa käyttöliittymän toiminnoista on kuitenkin tässä vaiheessa kytketty pois päältä tarpeettomina.

Käyttäjän ollessa tyytyväinen tiedoston sisältöön voi hän tallentaa konfigurointitiedoston painamalla save file -näppäintä. Save file -näppäintä painettaessa avautuu käyttäjälle dialogi-ikkuna, josta hän voi nimetä konfigurointitiedoston ja valita mihin sijaintiin se tallennetaan.

7.3 Valikkorakenne

Sovelluksen käynnistyessä avautuu ohjelman oletusnäkyvä. Sovelluksen yläreunassa on alavetovalikot, joista löytyy kaikki käyttöliittymän asetukset ja tärkeimmät tiedot sovelluksesta. Näitä valikoita on tarkoitus käyttää lähinnä erikoistapauksissa jos ei muuten löydä tarvittavia käyttöliittymän hallintanäppäimiä tai jos haluaa hakea tarkempaa tietoa sovelluksesta ohje-valikosta.

Yläreunan alavetovalikon alapuolella on eri kenttiä, joista löytyy käyttöliittymän tärkeimmät ohjaukset. Siitä löytyy eri näppäimiä ja teksti- sekä valintakenttiä. Vasemmassa reunassa olevat kontrollit vaikuttavat koko konfigurointitiedoston sisältöön, keskellä olevat kontrollit ovat symbolitauluille ja data blockeille sekä oikeassa reunassa kontrollit muunnokselle ja konfigurointitiedoston tallennukselle. Kaikki hallintapainikkeet on aseteltu sellaiseen järjestykseen, joka ohjaa käyttäjää aloittamaan asetusten tekemistä loogisessa järjestyksessä siten, että vaiheita ei jää tekemättä. Valikoissa on tarkoitus liikkua ylhäältä alas ja vasemmalta oikealle. Jos käyttäjä kuitenkin yrittää tehdä asioita väärässä järjestyksessä, ohjaa ohjelma käyttäjää tekemään ensin tarvittavat toiminnot. Käytössä olevat kontrollit näkyvät aktiivisina ja toimivat normaalisti. Kontrollit jotka eivät ole kyseisessä ohjelman vaiheessa toiminnassa näkyvät ei-aktiivisina, ja niitä painaessa ei myöskään tule visuaalista indikointia kontrollin toiminnasta. Tämä varmistaa käyttäjän ymmärtävän selvästi, mitkä kontrollit hänellä on käytettävissään.

Sovelluksen oletusnäkyvän keskellä olevan taulukon kontrollit olisivat olleet työläitä käyttää pitkien etäisyyksien vuoksi, jos ne olisi asetettu erillisiksi painikkeiksi valikkoihin. Taulukon kontrolleille päätettiin tehdä oma valikko hiiren oikean näppäimen alle, jolloin hiiren oikeata näppäintä painettaessa tulee esiin valikko, josta pääsee suoraan käsiksi kontrolleihin ja samalla hiiren liikkeitä ja käyttäjän katseen liikkeitä pysyvät minimissään.

7.4 Konfigurointitiedoston käyttö

Crimson 3 -konfigurointiohjelmaan voidaan tuoda kahden tyyppisiä tiedostoja, Unicode Text Files- ja ANSI CSV Files -tyyppisiä tiedostoja. Simred-sovellus käyttää ANSI CSV Files -tyyppistä tiedostoa. Simred-konfigurointitiedostoa voidaan hyödyntää niin uuteen kuin vanhaankin Crimson 3 -projektiin. Konfigurointitiedoston avulla voidaan esimerkiksi helpottaa projektien yhdistämistä toteutuksella, jossa päivitetään vanha Siemens-toteutusta luomalla uusi Red Lion -toteutus vanhan Siemens PLC -toteutuksen päälle tai toteutuksella, jossa yhdistetään vanha Red Lion -toteutus ja vanha Siemens -ratkaisu. Simred-sovelluksen konfigurointitiedosto voi tulla tarpeeseen myös uudessa vasta tekeillä olevassa projektissa.

Simred-konfigurointitiedoston käyttöön ottaminen Crimson 3 -konfigurointiohjelmassa on helppoa. Crimson 3 -konfigurointiohjelman import-näppäimellä avataan konfigurointitiedoston sijainti ja valitaan tiedosto tyyppiä ANSI CSV File. Crimson 3 -ohjelman dialogi-ikkunassa tulee kaikki sijainnin ANSI CSV File -tyyppiset tiedostot näkyviin, joista valitaan haluttu tiedosto. Valittaessa haluttu tiedosto, lisää Crimson 3 kaikki konfigurointitiedoston tiedot suoraan olemassa olevaan projektiin säilyttäen vanhan projektin muuten ennallaan.

8 Käyttöliittymän suunnittelu

Hyvän käyttöliittymän pitäisi pyrkiä minimoimaan seuraavia asioita:

- Älyllisen työn minimointi (yksinkertainen ja selkeä käyttö)
- Fyysisen työn minimointi (turhat painikkeiden painamiset ja valikoissa liikumiset pois)
- Muistinkäytön minimointi (monimutkaisuuden välttäminen)
- Visuaalisen rasittavuuden minimointi (olennainen esillä) [20, s.131.]

Hyvä käyttöliittymäsuunnittelu vaatii monien asioiden ymmärrystä ja niiden huomioon ottamista. Kun käyttöliittymän suunnittelussa hyödynnetään tieto ihmiselle luonteen omaisista tavoista nähdä asioita, ymmärtää ja ajatella, tulee valmiista käyttöliittymästä varmemmin helppokäyttöinen ja tehokas. Käyttöliittymää suunnitellessa tiedon visuaa-

linen esitystapa on hyvä pohtia tarkkaan, jotta pystytään tehostamaan käyttäjän tiedon vastaanottokykyä ja ymmärrystä. Väsymyksen ja muun rasituksen minimoimiseksi tulee ottaa huomioon myös käyttöliittymän ergonominen asettelu niin käden koordinaation kuin silmien liikkeiden helpottamisen näkökulmasta. Myös ohjelmistojen ja laitteiden aiheuttamat rajoitukset on hyvä ottaa huomioon. [20, s. xix - xx]

Hyvässä käyttöliittymässä on sopiva määrä tietoa kerralla näytöllä, sillä liian vähän on tehotonta ja liikaa on sekavaa. Toimintojen ja päätöksenteon kannalta olennainen ja tarpeellinen tieto tulisi näyttää yhdellä näytöllä, ei pirstaloituna usealle näytölle. Käyttöliittymässä esitetty tieto olisi myös hyvä jakaa osiin loogisesti, merkityksellisesti ja järkevästi. Käyttöliittymän elementit tulee asetella käyttäjän odottamiin sijainteihin ja kokonaisuuksiin sekä järjestää ne käyttäjän näkökulmasta tarpeelliseen ja loogiseen järjestykseen. Käyttöliittymän elementtien huolellisella asettelulla voidaan ohjata käyttäjän silmää ruudulla oikeisiin paikkoihin luonnollisin liikeradoin, jolloin minimoidaan käyttäjän osoittimen ja silmien liikettä, ja tehdään näin käyttöliittymän käyttökokemuksesta miellyttävä ja vaivaton. [20, s. 728.]

Visuaalisesti miellyttävä käyttöliittymä muodostuu tasapainoisen kokonaisuuden avulla, jossa on muun muassa sopivasti tyhjää tilaa ja oikein ryhmitetyt kuvakkeet. Käyttöliittymän eri elementit eivät erotu toisistaan selvästi, jos ruudut ovat liian täyteen ahdettuja. Huolitellulla tilan käytöllä ja elementtien ryhmyksillä voidaan korostaa käyttöliittymän tärkeitä alueita ja kohdistaa näin käyttäjän huomio toiminnan kannalta kriittisille alueille. [20, s.728 - 729.]

Käyttöliittymää luodessa omiin kokonaisuuksiinsa ryhmitellyt elementit tulee nimetä sopivin otsikoin ja tarvittaessa ala-otsikoin. Otsikoiden tulisi olla tarkoituksenmukaisia, ytimekkäitä ja niistä pitäisi selvittää kontrollien toiminta. Halutessasi voit tuoda eri ryhmiä paremmin esille käyttämällä kehyksiä. Erilaisilla tai eri paksuisilla kehyksillä voit myös tarvittaessa korostaa haluamiasi asioita entisestään. [20, s. 730.]

Käyttöliittymän samalla tasolla olevien elementtien muodostamien linjojen määrä tulisi pitää minimissään. Eri linjojen määrän vähentäminen yksinkertaistaa käyttöliittymää ja tekee siitä visuaalisesti miellyttävämmän. Käyttöliittymän rakenteesta tulee johdonmukaisempi ja tekee siitä käyttäjälle helpommin omaksuttavamman. [20, s. 738.]

Elementtien linjauksien ja elementtien ryhmittämisen tulisi auttaa käyttäjää suunnistamaan käyttöliittymässä. Tärkeimmät ja eniten käytetyimmät kontrollit tulisi sijoittaa ylös vasemmalle. Toimintojen yleisen suunnan tulisi silti säilyä ylhäältä alas, vasemmalta oikealle. Toisistaan riippuvien kontrollien tapauksessa, tulisi edellisen kontrollin olla seuraavan yläpuolella tai vasemmalla puolella. [20, s.728 - 729, s. 738.]

Suunnitellessa käyttöjärjestelmän väriteemaa tulisi ottaa huomioon ihmisen näköjärjestelmä ja se mitä ongelmia eri värit voivat aiheuttaa riippuen katseluympäristöstä. Huomioon tulee ottaa myös laitteisto jolla käyttöliittymää käytetään sekä tehtävä jota ollaan tekemässä. [20, s. 702 - 703.]

Värejä voi käyttää korostamaan asioita käyttöliittymässä tai kiinnittämään huomion haluttuun kohteeseen. Jos käyttöliittymästä löytyy monia samankaltaisia kontrolleja, voidaan toisiinsa liittyvät komponentit yhdistää värejä käyttämällä. Toisaalta värejä voi käyttää myös toisinpäin korostamaan käyttöliittymän komponenttien erottelua toisistaan. [20, s. 704.]

Värien käytössä tulisi olla johdonmukainen koko käyttöliittymän osalta. Jos käyttäjä yhdistää värin toimintoihin ajan myötä, niin yhdistää hän muutkin samanväriset elementit kyseisiin toimintoihin myös jatkossa. Jos värin tarkoitus jossain välissä muuttuu, aiheuttaa tämä sekavuutta ja väärinkäsityksiä [20, s. 713.]

Värien suunnittelussa on hyvä ottaa huomioon väreihin liitetyt tarkoitukset, jotta vältetään väärinymmärryksiltä ja ristiriidoilta. Esimerkiksi yleensä ihmiset odottavat punaisen tarkoittavan vaara tai pysähtymistä, keltaisen varoitusta, hiljentämistä tai testiä. Oikea värien käyttö vaatii huolellisuutta, kokemusta ja käyttäjien huomioon ottamista. [20, s. 707.]

Värejä tulee käyttää säästeliäästi, sillä värit kiinnittävät helposti huomion kohteeseen. Värien määrän tulisi riippua käyttötarkoituksesta; jos kaksi väriä täyttää tehtävänsä, käytä kahta väriä, ja jos kolme väriä sopii parhaiten käyttötarkoitukseen, niin käytä kolmea väriä. Parhaaseen erottelukyvyn ruudulla päästään kuitenkin käyttämällä enimmillään neljästä viiteen väriä läpi värispektrin. [20, s. 702 - 704.]

9 Simred-sovelluksen käyttöliittymän suunnittelu ja toteutus

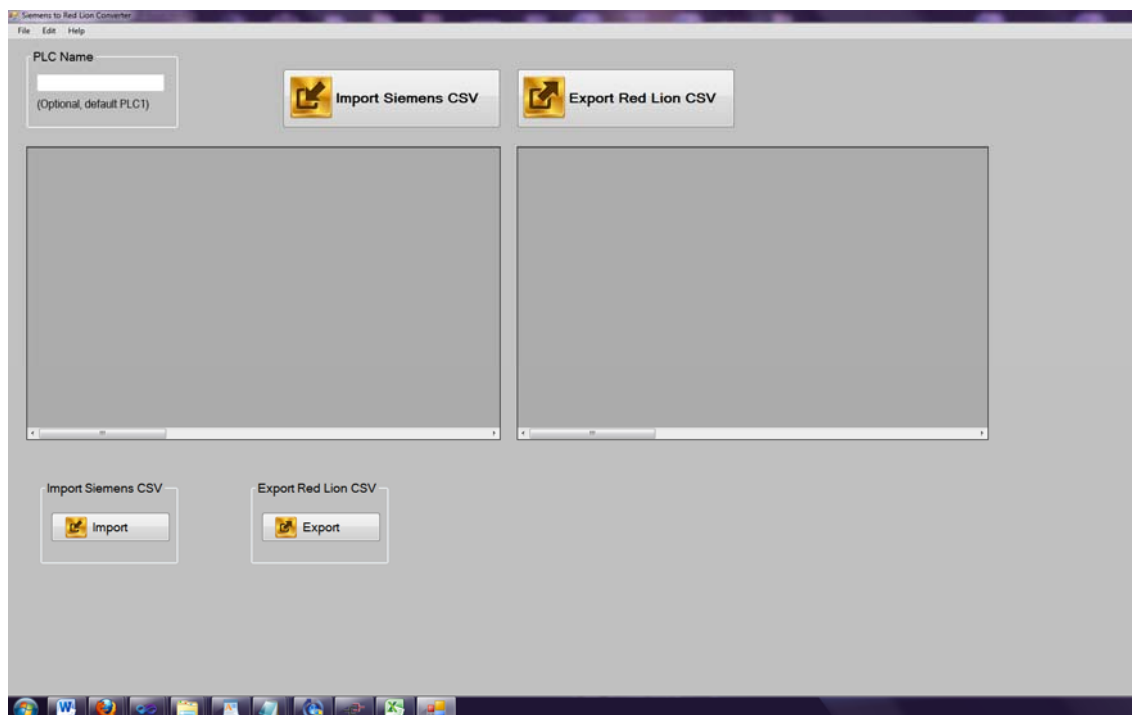
Simred-sovelluksen käyttöliittymä tehtiin käyttämällä Visual Studio 2010 - käyttöliittymätyökaluja. Käyttöliittymää rakennettiin koko ajan ohjelmointityön ohella ja sovelluksen kehittyessä myös käyttöliittymää vietiin samaan suuntaan. Näin varmistettiin käyttöliittymän toimivuus ohjelman kehittyessä ja muuttuessa.

9.1 Käyttöliittymän kehitys

Sovelluskehityksessä tehdyn suunnittelun perusteella, tiedettiin jo hyvin pitkälle mitä kontrolleja näytöllä tulisi olemaan. Tämä tieto yhdistettynä vaatimusmäärittelyyn antoi jo varsin hyvän pohjan käyttöliittymän suunnitteluun. Näiden tietojen perusteella pystyttiin tekemään päätökset väriteemasta ja pystyttiin toteuttamaan ensimmäinen versio käyttöliittymän ulkonäöstä. Jo suunnittelusta lähtien pidettiin hyvän käyttöliittymän ominaisuudet mielessä ja yritettiin lähteä kehittämään käyttöliittymää oikeaan suuntaan.

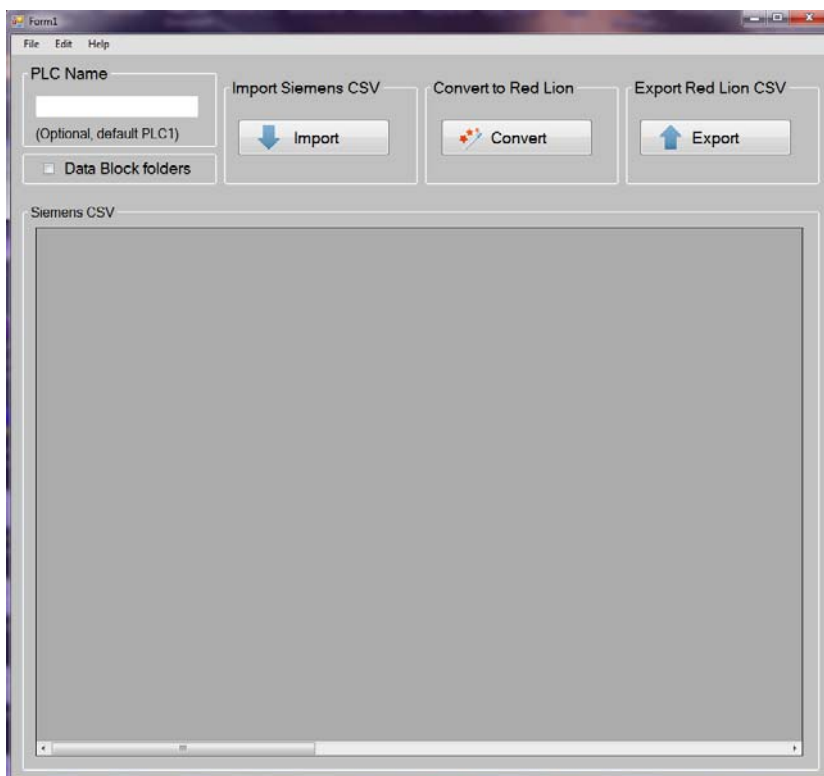
Alkuvaiheen suunnitteluun kuului myös sopivien graafisten elementtien etsiminen käyttöliittymään. Omien graafisten elementtien tekeminen olisi kestänyt liian kauan, joten ne päätettiin etsiä muualta. Graafisten elementtien etsiminen oli varsin työlästä, sillä niiden ei haluttu rikkovan kenenkään tekijänoikeuksia vaan niiden tuli olla vapaata käyttää missä tahansa sovelluksissa. Graafisten elementtien tuli myös kuvastaa hyvin tarkoitusta jossa sitä käytetään.

Alkuvaiheessa käyttöliittymän tekeminen oli lähinnä kokeilua ja eri toteutuksien toimivuuden tutkimista. Kuvassa 8. näkyy käyttöliittymän yksi alkuvaiheen versioista. Kuvassa näkyy käyttöliittymän kokeilu, jossa tutkitaan elementtien kokoja sekä ikonien sopivuutta. Tässä vaiheessa käyttöliittymä oli vielä suunniteltu toimivaksi kahden taulukon avulla.

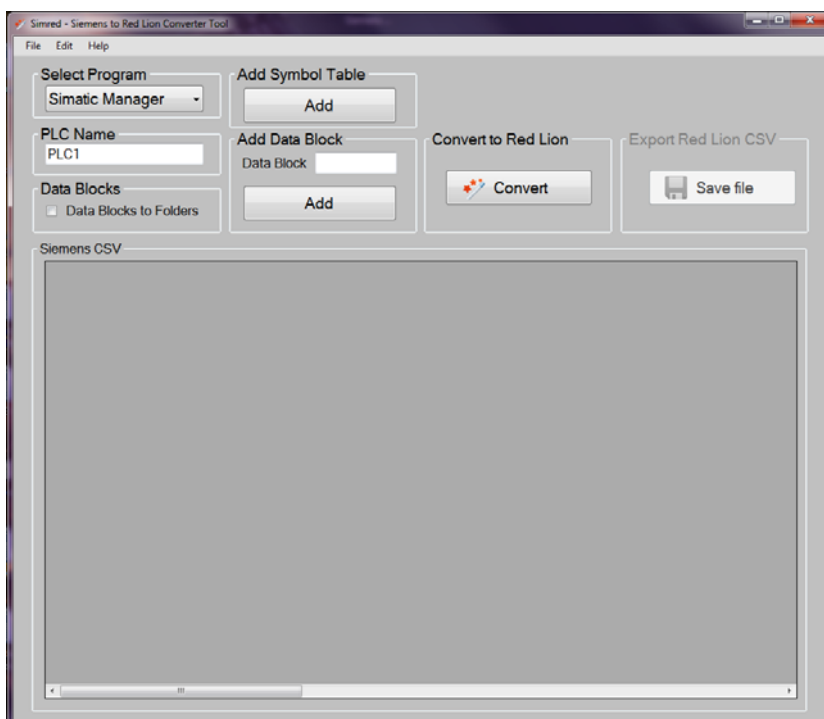


Kuva 8. Simred-sovelluksen käyttöliittymä suunnitteluvaiheessa.

Käyttöliittymän kehitystä jatkettiin samaan kehitys suuntaan tästä eteenpäin. Käyttöliittymä kuitenkin siirtyi yksi taulukkoiseen tietojenkäsittely- ja tietojen selausjärjestelmään. Ikonien tyyli vaihtui myös moneen kertaan. Testissä oli monia eri kuvake ja painike ratkaisuja. Käytössä oli muun muassa Microsoft Office -tyylinen ratkaisu, jossa löytyy pikku kuvakkeet vasemmasta yläreunasta. Kuvassa 9 oleva käyttöliittymä on kuitenkin jo lähellä lopullista käyttöliittymää, mutta kontroleja jää tästä versiosta vielä puuttumaan. Kehitysvaiheessa olevassa käyttöliittymässä näkyvä import-painikkeen toiminto on kuitenkin valmiissa käyttöliittymässä viety ruudun yläreunassa olevan file-valikon taakse, koska import-painiketta käytetään harvemmin. Kuvassa 10 on vielä vertailuna valmis käyttöliittymä.



Kuva 9. Kehitysvaiheessa olevan Simred-sovelluksen käyttöliittymä.



Kuva 10. Valmiin Simred-sovelluksen käyttöliittymä.

9.2 Käyttöliittymän ulkoasu

Käyttöliittymän ulkoasun haluttiin olevan yksinkertainen ja selkeä. Aloitusnäkyvän oli tarkoitus ohjata käyttäjä heti tekemään haluamiaan asioita. Oletusnäkyvässä haluttiin näkyvän juuri ne tärkeimmät valinnat, joita käyttäjän tarvitsee tehdä valitessaan konfigurointitiedostoon lisättäviä asioita. Käyttäjää ei haluttu hämmentää liian monilla valinnoilla, vaan toiminnan haluttiin olevan suoraviivaista, joten näytöllä näkyvien elementtien määrä pidettiin vähäisenä. Kaikki ylimääräiset asetukset jätettiin sovelluksen yläreunassa oleviin alasvetovalikkoihin. Käyttäjän valitessa konfigurointitiedoston tietoja haluttiin hänellä olevan käytettävissä vain ne työkalut, joita ohjelman siinä vaiheessa pystyi käyttämään.

Ohjelmaan on tarkoitus tuoda paljon tietoja suoraan Siemens projektista siinä muodossa kuin ne siellä ovat, joten käyttäjän haluttiin koko ajan näkevän, mitä tietoja hän on lisäämässä. Käyttäjän oli tärkeä myös tarvittaessa päästä käsittelemään ja muokkaamaan jo valitsemiaan tietoja, joten keskellä oletusnäkyvää keskeisimmäksi osaksi käyttöliittymää haluttiin taulukkonäkymä, jossa näkyisi omissa soluissaan kaikki tiedot, joita konfigurointitiedostoon oltiin tuomassa.

Ulkoasun haluttiin olevan ajattoman yksinkertainen ilman suurempia graafisia hienouksia. Toivomusten perusteella sopivaksi väriteemaksi ajateltiin rauhallisia harmaita sävyjä sekä mustaa ja valkoista. Valintaan vaikutti myös mustan, eri harmaiden sävyjen ja valkoisen tarjoama hienojakoinen erottelukyky, jota pystytään hyödyntämään Simred-sovelluksen käyttöliittymän sisällön puolesta erittäin hyvin. Kuvia ohjelmassa käytettiin painikkeiden logoina selkeyttämään näkymää, esimerkiksi tiedoston tallennus painikkeelle käytettiin perinteistä sinistä disketin kuvaa, joka on hyvin käytetty laajasti eri Windows-ohjelmissa ja siten kuvastaa jo itsessään, mitä painike tekee. [20, s. 711.]

10 Päätelmät

Projektin lopputuloksena saatiin toimiva työkalu Siemensin ohjelmoitavilla logiikoilla toteutettujen projektien ja Red Lionin tuotteilla tehtyjen toteutusten yhdistämiseen. Simred-sovelluksesta tuli vaatimusmäärittelyiden mukainen ohjelma ja projekti saatiin Sarlin Oy Ab:n vaatimiin aikamääreisiin mennessä valmiiksi.

Visual Studio -ohjelmistoalusta ja C#-ohjelmointikieli osoittautuivat hyväksi ja toimiviksi valinnoiksi. Visual Studio -ohjelmistoalusta osoittautui lähes korvaamattomaksi avuksi. Visual Studion valmiiksi rakennettujen osien hyödyntäminen nopeutti ja helpotti ohjelmointityötä huomattavasti. Visual Studion käyttöliittymätyökalujen avulla oli helppo suunnitella käyttöliittymän toteutus. Myös C#-ohjelmointikielen tehokkuus nopeutti ohjelmoinnin etenemistä ja se auttoi vähentämään ohjelmointivirheiden määrää. Tämä auttoi erityisesti ohjelman testausvaiheessa, koska siellä ei enää tarvinnut tehdä kovin paljoa virheiden korjauksia. Pienessä yhden ohjelmoijan ohjelmistonkehitysprojektissa Visual Studio -ohjelmistoalustan käytön tehokkuus varmasti korostui vielä entisestään verrattuna suurempien projektiryhmien toteutuksiin.

Projektin eri vaiheiden huolellinen suunnittelu säästi mietintäaikaa projektin toteutusvaiheessa ja maksoikin itsensä varmasti takaisin. Suunnittelussa olisi silti pitänyt käyttää vielä enemmän aikaa koodin optimointiin, sillä siihen jäi vielä parannettavaa. Ohjelma on kuitenkin verrattain kevyt, joten tässä ohjelmointiprojektissa koodin optimointi ei kuitenkaan ole yhtä oleellista kuin monissa muissa ohjelmointiprojekteissa.

Eniten aikaa projektissa kului käytännön ohjelmistotyön tekemiseen, joka vei vähän yli puolet työhön käytetystä ajasta. Suunnittelu ja toteutuksiin perehtyminen vei noin 25 % käytetystä ajasta ja käyttöliittymä vei noin 15 % käytetystä ajasta.

Projektia voisi vielä jatkaa keräämällä laajemmalla asiakaskunnalta tietoa heidän käyttökokemuksistaan ja mahdollisista lisäominaisuuksista, joita he vielä tarvitsisivat sovellukseen. Tietojen pohjalta sovellusta voitaisiin kehittää eteenpäin ja lisäominaisuuksilla ohjelmasta saataisiin vielä monipuolisempi, mikäli lisäominaisuuksille ilmenee tarvetta.

Lähteet

1. Sarlin kalvosarja 2014. PDF-dokumentti. Sisäinen materiaali
2. Red Lion HMI FIN. PDF-dokumentti. Sisäinen materiaali
3. Red Lion G3 Panels. Verkkodokumentti.
<http://www.redlion.net/Products/HumanMachineInterface/ProductFamilies/G3Panels.html> Viitattu: 29.3.2014
4. Red Lion G3 Panels. Verkkodokumentti.
http://www.redlion.net/News/G3_brochure_web.pdf Viitattu: 29.3.2014
5. Red Lion Kadet Series HMIs. Verkkodokumentti.
<http://www.redlion.net/aboutus/news/news2008/g3kadetserieshmis.html> Viitattu: 29.3.2014
6. Red Lion Data Station Plus. Verkkodokumentti.
<http://www.redlion.net/products/humanmachineinterface/datastationplus.html> Viitattu: 29.3.2014
7. Red Lion Data Station Plus. Verkkodokumentti. <http://www.redlion.net/News/DSP-Brochure.pdf> Viitattu: 29.3.2014
8. Ohjelmoitavat logiikat (SIMATIC). Verkkodokumentti.
http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic.php Viitattu: 30.3.2014
9. Simatic Manager ohjelman sisäinen materiaali: Introduction 30.3.2014
10. Lista STEP 7 Professional V11 ja WinCC Professional V11 tukemista moduuleista. Verkkodokumentti.
https://a248.e.akamai.net/cache.automation.siemens.com/dnl/jM/jMxOTk3AAAA_50029532_FAQ/STEP7_V11_Support_HW_List.pdf Viitattu: 4.4.2014
11. Introduction to the SIMATIC Manager. Verkkodokumentti.
http://ipnet.agh.edu.pl/Materials1/Module2/SIMATIC_MANAGER.pdf Viitattu: 4.4.2014
12. Siemens SIMATIC S7-300. Verkkodokumentti.
http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic/s7_300.php Viitattu: 4.4.2014

13. TIA Portal -ohjelmointityökalu. Verkkodokumentti.
http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/tia_portal.php Viitattu: 4.4.2014
14. SIMATIC StartDrive -ohjelmisto. Verkkodokumentti.
http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/kayttotekniikka_ja_liikkeenohjaus/liikkeenohjausjarjestelmat/sinamics_drive_solutions/sinamics_startdrive.htm Viitattu:4.4.2014
15. Totally Integrated Automation Portal (TIA -Portal). Verkkodokumentti.
<https://www.industry.siemens.com/topics/global/en/tia-portal/Pages/default.aspx> Viitattu: 4.4.2014
16. Haikala, Ilkka., Mikkonen, Tommi. 2011. Ohjelmistotuotannon käytännöt. Talentum Media Oy.
17. Rahikainen, Markku. 2011. C# Windows-ohjelmointi - Software Development Demystified. Stickmansoft. Juvenes Print - Tampereen Yliopisto Oy.
18. Moghadampour, Ghodrat. 2012. C#-ohjelmointi. Sanoma Pro Oy.
19. Hejsberg, Anders., Torgersen, Mads., Wiltamuth, Scott., Golde, Peter. 2011. The C# Programming Language -Fourth Edition. Microsoft Corporation.
20. Galitz, Wilbert. 2007. The Essential Guide to User Interface Design - An Introduction to GUI Design Principles and Techniques. Wiley Publishing Inc.