

Antti Ruohio

# Rob-Ex-tuotannonsuunnittelujärjestelmän integrointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikka

Insinööriytyö

23.5.2013

Tekijä(t) Otsikko	Antti Ruohio Rob-Ex-tuotannonsuunnittelujärjestelmän integrointi
Sivumäärä Aika	44 sivua + 3 liitettä 23.5.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	automaatiotekniikka
Suuntautumisvaihtoehto	kappaletavara-automaatio
Ohjaaja(t)	lehtori Jukka-Pekka Pirinen lehtori Antti Liljaniemi
<p>Tämän insinööriyön aiheena on Rob-Ex-tuotannonsuunnittelujärjestelmän integroinnin toteuttaminen automaatiojärjestelmän kanssa sekä erilaisten integrointimenetelmien ja niihin liittyvien käsitteiden sekä standardien tutkiminen. Insinööriyö toteutettiin osana Metropolia Ammattikorkeakoulun automaatiotekniikan koulutusohjelman toimeksiantona suoritettavaa projektia, jossa automaatiolaboratorioon toteutettiin uudenlainen tuotantojärjestelmä koulutuskäyttöön.</p> <p>Opinnäytetyössä keskitytään tarkastelemaan kirjallisuuden, konsultoinnin sekä sähköisten dokumenttien ja ohjeiden pohjalta Rob-Ex-tuotannonsuunnittelujärjestelmän erilaisia mahdollisia integrointitekniikoita ja niiden toimintaperiaatteita sekä niihin liittyviä käsitteitä. Lisäksi selvitetään miten ISA-95 -tiedonsiirtostandardi liittyy toteutukseen. Myös Rob-Ex:n tuotannonsuunnitteluun liittyviä ohjelmallisia ominaisuuksia ja niiden toiminnallisuutta sekä toiminnan- ja tuotannonohjausjärjestelmän toimintaperiaatteita käsitellään.</p> <p>Toteutukseen sisältyi teoriaosuuden lisäksi Rob-Ex-tuotannonsuunnittelujärjestelmän konfigurointi automaatiolaboratorion laitteistoa vastaavaksi ja järjestelmän integrointi automaatioon, toteuttamalla laitteiston mallinnus sekä tiedonsiirto laitteistoa ohjaavan valvomosovelluksen ja tuotannonsuunnittelujärjestelmän välillä. Tiedonsiirron toteuttaminen sisälsi sovelluskehitystä, jonka tuloksena syntyi sovellusmuotoinen työkalu tiedonvälitystä varten.</p> <p>Lopputyössä käydään myös läpi tiedonvälitystyökalun toimintaa ja ohjelmointiprosessia. Lisäksi analysoidaan tarkemmin toteutettavaksi valittua tiedonsiirtomenetelmää sen soveltuvuuden ja ominaisuuksien kannalta. Lopuksi vielä pohditaan, miten järjestelmää voi kehittää kattavammaksi ja edistyneemmäksi tulevaisuudessa, sekä tarkastellaan projektin onnistumista kokonaisuutena.</p> <p>Työn tuloksena saavutettiin toimiva integraatio Rob-Ex-tuotannonsuunnittelujärjestelmän ja automaatiolaitteiston välillä, jota on mahdollista laajentaa tulevaisuudessa useiden automaatiolaitteistojen käytettäväksi.</p>	
Avainsanat	ROB-EX, ISA-95, Valmistuksenohjaus, Integrointi, Automaatio, MES, ERP, XML, CSV, OPC, SQL, Webservice

Author(s) Title	Antti Ruohio Integration of Rob-Ex-Manufacturing Planning System
Number of Pages Date	44 pages + 3 appendices 23 May 2013
Degree	Bachelor of Engineering
Degree Programme	Automation Engineering
Specialisation option	Manufacturing Automation
Instructor(s)	Jukka-Pekka Pirinen, Senior Lecturer Antti Liljaniemi, Senior Lecturer
<p>The purpose of this thesis was to integrate the Rob-Ex manufacturing planning system with an automation system and conduct research about different methods used for integration between systems, including integration related concepts and standards. This thesis was executed as part of a project being carried out by the Metropolia University of Applied Sciences Automation Technology degree program. As a result of the project a new production system was created in the automation laboratory for educational purposes.</p> <p>This thesis focuses on different possible methods, principles and concepts for integrating the Rob-Ex manufacturing planning system with research basing on literature, consulting, electronic documents and instruction manuals. In addition the relation between the ISA-95-standard and the realization of the project is being examined. Also the functionality of different programmatic features included in the Rob-Ex and the general principles of enterprise resource planning and manufacturing execution systems are being covered.</p> <p>In addition to theoretical research, the implementation of this thesis included the configuration of the Rob-Ex manufacturing planning system to meet the requirements of the physical equipment utilized in the automation laboratory and the integration with the automation system itself. This was achieved by creating a model of the equipment into Rob-Ex and data exchange between Rob-Ex and the automation system. The creation of data exchange included software development resulting in a data transfer application.</p> <p>The functionalities and the programming process of the data transfer application are also being reviewed in this thesis. In addition the data transfer method chosen for the project is more closely inspected according to its compatibility and different features. In conclusion the development possibilities and the success of the project as a whole are estimated.</p> <p>As a result of the project a functional integration between Rob-Ex and the automation system was established including the option to expand the system to be utilized with multiple different automation systems.</p>	
Keywords	ROB-EX, Manufacturing Execution System, Integration, Automation, MES, ERP, XML, CSV, OPC, SQL, Webservice

## Sisällys

Tiivistelmä

Abstract

Lyhenteet

1	Johdanto	1
1.1	Taustaa	1
1.2	Alkutilanne	1
1.3	Tavoite	1
1.4	Tutkimusmenetelmät	2
2	Projektin kuvaus	3
2.1	Kuvaus kokonaisprojektista	3
2.2	Kuvaus omasta osuudesta	4
2.3	Projektin päämäärä	5
2.4	Projektin toimintasuunnitelma	6
3	Käsitteistä	7
3.1	ISA-95 -malli	7
3.1.1	ISA-95 -standardi	7
3.1.2	ISA-95 -rakenne	8
3.2	ERP -järjestelmät	10
3.2.1	ERP	10
3.2.2	Toimintaperiaate	10
3.2.3	Etuja ja haittoja	11
3.3	MES -järjestelmät	12
3.3.1	MES	12
3.3.2	Rakenne & Toimintaperiaate	12
3.3.3	Toiminnallisuudet	13
3.3.4	MES -järjestelmien tarjoamat edut	14
3.4	Automaatio	14
3.4.1	Määritelmä	14
3.4.2	Automaation lajit	14
3.4.3	Automaation käyttö	15
4	Erilaiset tiedonsiirto ja integrointimahdollisuudet	16
4.1	SQL -tietokannat	16

4.1.1	SQL -perusta	16
4.1.2	Toimintaperiaate ja käyttö	16
4.2	ODBC ja SQL	16
4.3	Webservice -tekniikka	17
4.4	OPC -tekniikka	17
4.4.1	OPC -standardit	17
4.4.2	OPC -toimintaperiaate	18
4.4.3	OPC -tarkoitus	19
4.5	XML	20
4.5.1	XML -standardi	20
4.5.2	XML -muotoiset dokumentit	20
4.6	CSV -tiedostomuoto	21
5	Rob-Ex	22
5.1	Ominaisuudet	22
5.2	Toimintaperiaate	23
5.3	Sovelluskohteet	24
5.4	Integrointimahdollisuudet	25
6	Toteutus	26
6.1	Valittu integrointitekniikka	26
6.2	Toteutustapa	26
6.3	Tietojen välitys	26
6.4	Toteutuksen eri vaiheet	28
6.4.1	Aloitus	28
6.4.2	Rob-Ex	28
6.4.3	Tiedonsiirtotyökalu	28
6.4.4	Raportointi	29
7	Tiedonsiirtotyökalu	30
7.1	Työkalun toimintaperiaate	30
7.2	Ominaisuudet ja käyttö	31
7.3	Ohjelmointi	32
7.3.1	Ohjelmointikieli ja valmistelut	32
7.3.2	Ohjelmoinnin toteutus	33
7.3.3	Käytetyt menetelmät	34
7.4	Ohjelmarakenne	39
7.5	Integrointi Rob-Ex:n ja WinCC:n välillä	41

7.6	Testaus	42
8	Yhteenveto	43
8.1	Aikaansaatu kokonaisuus	43
8.2	Vertailu tavoitteisiin	43
8.3	Kehitysmahdollisuudet tulevaisuudessa	43

Lähteet

Liitteet:

Liite 1. ReadWriter for Rob-Ex v1.0 -tiedonvälitystyökalun käyttöohje

Liite 2. ReadWriter for Rob-Ex v1.0 -tiedonvälitystyökalun ohjelmakoodi

Liite 3. Tuotantoautomaatio - Laboratorioharjoitus / Rob-Ex tuotannonsuunnittelujärjestelmä

## Lyhenteet

ANSI	American National Standards Institute. Yhdysvaltalainen voittoa tavoittelematon standardien kehittämistä valvoma organisaatio.
ASCII	American Standard Code for Information Interchange. 7-bittinen tietokoneiden käyttämä merkistö, joka sisältää amerikanenglannissa käytettävät kirjaimet, sekä tietyn määrän erikoismerkkejä.
B2M	Business to Manufacturing. Olennaisten tietojen välittäminen yritystoiminnasta valmistustoiminnalle.
BOM	Byte Order Mark. Unicode merkki, joka ilmaisee tekstitiedoston tavujärjestyksen.
CIM	Computer Integrated Manufacturing. Tietokoneiden käyttö tuotannonohjauksessa.
CSV	Comma Separated Values. Tekstitiedosto, jossa tiedoston sisältämät arvot ovat eroteltuina pilkuilla.
DTD	Document Type Definition. SGML- ja XML-kielten yhteydessä käytetty rakennemäärittely.
ERP	Enterprise Resource Planning. Tietojärjestelmä, joka integroi yrityksen eri toimintoja.
HTTP	Hypertext Transfer Protocol. Tiedonsiirtoprotokolla, jota käytetään tietojen välityksessä Internetissä.
IANA	Internet Assigned Numbers Authority. Järjestö, joka valvoo esimerkiksi IP- ja www-osoitteiden jakamista Internetissä.
IETF	Internet Engineering Task Force. Internet protokollien standardoinnista vastaava organisaatio.

ISA	Instrumentation Systems and Automation Society. Kansainvälinen voittoa tavoittelematon järjestö, joka pyrkii tukemaan automaation kehitystä.
ISA95	Enterprise Control Integration Standard. Yrityksen eri osien välistä integraatiota määrittelevä standardi.
LAN	Local Area Network. Paikallinen tietoverkko eli lähiverkko.
MES	Manufacturing Execution System. Valmistuksenohjausjärjestelmä, jonka avulla tuotantoa voidaan ohjata ja analysoida.
ODBC	Open Database Connectivity. Tietokantojen välinen avoin rajapinta.
OEE	Overall Equipment Efficiency. Analyysimenetelmä, jonka avulla tuotantoa pyritään tehostamaan.
OLE	Object Linking and Embedding. Teknologia, jonka avulla tietorakenteita voidaan upottaa ja linkittää.
OPC	Open Connectivity via Open Standards. Rajapinta, jonka avulla pyritään mahdollistamaan erilaisten toimilaitteiden yhteensopivuus.
RFC	Request For Comments. IETF:n julkaisemia Internet-standardeja.
SGML	Standard Generalized Markup Language. Metaohjelmointikieli, jonka avulla voidaan määritellä dokumenttien merkintäkieliä.
SOAP	Simple Object Access Protocol. Tiedonsiirtoprotokolla, jonka avulla tietoa voidaan siirtää verkon ylitse XML-muotoisena.
SQL	Structured Query Language. Kyselykieli, jonka avulla relaatiotietokantoja voidaan käyttää.
UDDI	Universal Description, Discovery and Integration. Yleinen ja avoin palvelurekisteristandardi, johon on listattu yrityksiä ja Webservice-palveluita.



UTF-8	8bit Unicode Standard. 8-bittinen tietokoneiden käyttämä Unicode-merkistö.
W3C	World Wide Web Consortium. Kansainvälinen yritysten ja yhteisöjen perustama järjestö, joka kehittää ja ylläpitää WWW-standardeja.
WSDL	Web Services Description Language. W3C:n kehittämä kieli joka kuvaa verkossa olevaa Webservice-toimintoa.
XML	eXtensible Markup Language. Tiedon merkintäkieli, jossa tiedon merkitys on kuvattavissa tiedon sekaan.

# 1 Johdanto

## 1.1 Taustaa

Ohjelmistopohjaisia tuotannonohjausjärjestelmiä pyritään hyödyntämään teollisuudessa pääosin tuotannon tehostamiseen. Keinoja toteuttaa tämä ovat hyvä ja joustava tuotannon suunnittelu sekä sen tehokas ja dynaaminen ohjaaminen. Tuotannon tehostaminen ja optimointi on oleellista monille yrityksille kannattavuuden ja sitä kautta kilpailukyvyn säilyttämisen kannalta. [2]

Metropolia Ammattikorkeakoulun automaatiotekniikan koulutusohjelma haluaa olla mukana kouluttamassa osaamista tuotannon- ja valmistuksenohjauksen saralla. Tämä loppuyö liittyy tarjottavan koulutuksen monipuolistamiseen ja sitä kautta laajemman osaamisen saavuttamiseen automaatiotekniikan koulutusohjelman tuloksena syksystä 2013 lähtien.

## 1.2 Alkutilanne

Projekti toteutetaan Metropolia Ammattikorkeakoulun Myyrmäen toimipisteessä sijaitsevassa automaatiolaboratoriossa. Laboratoriossa on projektiin liittyen valmiina fyysinen tuotantolaitteisto sekä projektin toteuttamiseen tarvittavat ohjelmistot ja työvälineinä toimivat PC-koneet. Tätä insinööriyötä koskien ohjelmistona käytettävä Rob-Ex Gantt Scheduler -tuotannosuunnittelujärjestelmä on valmiiksi asennettuna verkkolevylle serverikoneelle, josta käsin sitä voidaan käyttää useammalla tietokoneella. Tässä tapauksessa projektin käytännön työt toteutetaan serverikoneella, jotta oikeat konfiguraatiot ohjelmistolle saadaan tallennettua keskitetysti. Ohjelmiston kehittäjänä on Novotek Oy, josta projektin aikana saadaan mahdollisesti konsultointiapua Rob-Ex-ohjelmiston konfigurointiin, sen eri ominaisuuksiin sekä käyttöön liittyen.

## 1.3 Tavoite

Insinööriyön tavoitteena on selvittää, miten ja minkälaisia integrointimenetelmiä on mahdollista käyttää Rob-Ex-tuotannosuunnittelujärjestelmää tehdastason automaati-

on kanssa integroitaessa. Vastausta haetaan myös siihen, mikä integrointimenetelmä on toteutuksensa suhteen edullisin ja soveltuu parhaiten automaatiotekniikan koulutusohjelman tarpeisiin. Lisäksi tavoitteena on selvittää, miten eri tiedonsiirtotekniikat toimivat ja miten niitä koskevat standardit tekniikoiden käyttämiseen liittyvät.

Käytännön tavoitteena tämän opinnäytetyön osalta on luoda toimiva opetuskäyttöön suunnattu järjestelmä, jonka avulla tuotannon suunnittelu on mahdollista ja tuotantotietoa voidaan välittää Rob-Ex-tuotannosuunnittelujärjestelmän sekä tehdastason automaatiota ohjaavan valvomosovelluksen välillä. Lisäksi tulevaisuuden kannalta on tavoitteena selvittää, minkälaisia kehitysmahdollisuuksia järjestelmän osalta on mahdollista toteuttaa, mikäli jatkokehitys ja järjestelmän laajentaminen tulevat myöhemmin aiheellisiksi.

#### 1.4 Tutkimusmenetelmät

Tutkimusmenetelminä tässä lopputyössä on tarkoitus soveltaa teoriapohjaista tutkimusta mm. kirjallisuuden sekä muun materiaalin, kuten sähköisten dokumenttien ja ohjeiden pohjalta. Näillä tavoilla lopputyössä tutkitaan mm. eri käsitteiden teoriaa ja tekniikoiden toimintaperiaatteita sekä ohjelmistojen ominaisuuksia.

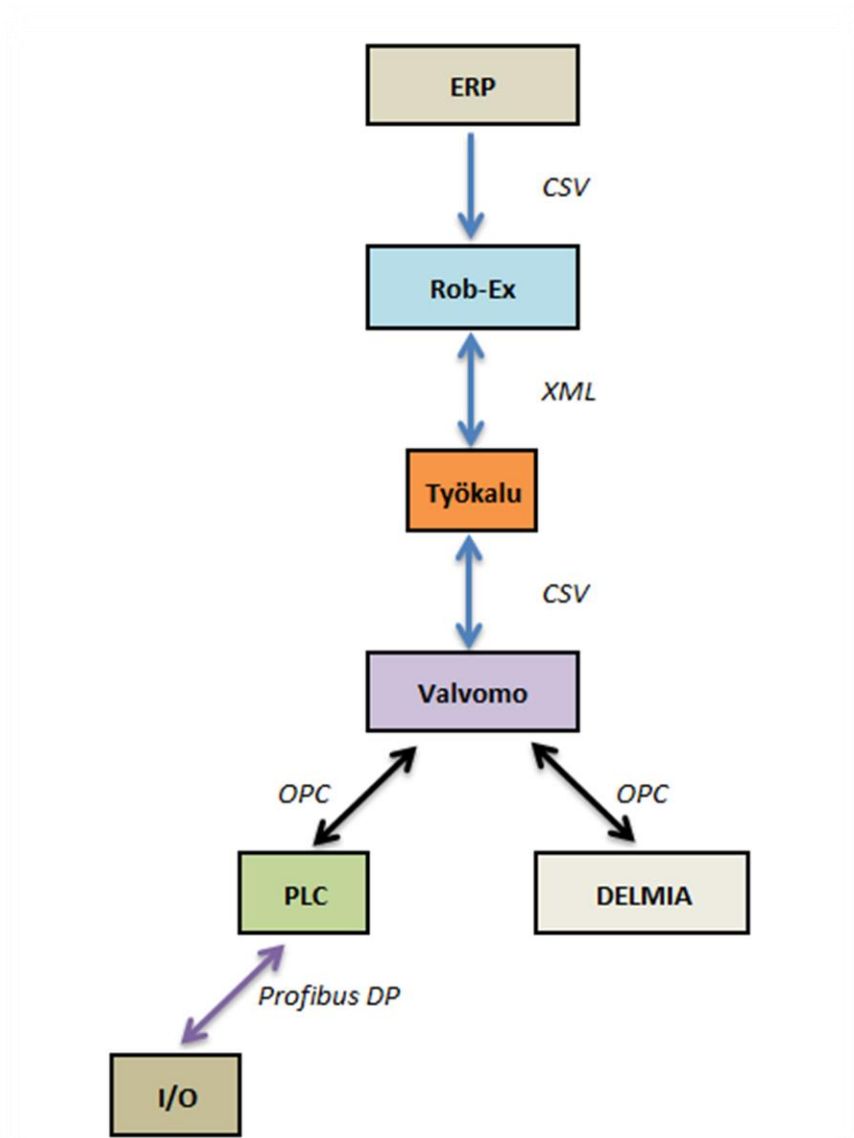
Muita tutkimusmenetelmiä olivat ohjelmistoilla suoritettut käytännön kokeet sovelluskehityksen sekä ohjelmistojen käytön merkeissä ja yhteyshenkilönä Metropolia Ammattikorkeakoulun kanssa toimivan Novotek Oy:n Rob-Ex-asiantuntijan Harri Ahlrothin haastattelu ja hänen tarjoamansa konsultointiapu koskien Rob-Ex-ohjelmiston ominaisuuksia ja integraatiomenetelmiä.

## 2 Projektin kuvaus

### 2.1 Kuvaus kokonaisprojektista

Opinnäytetöinä Metropolia Ammattikorkeakoulun automaatiolaboratorioon toteutettavaan tuotantojärjestelmään sisältyy yhteensä kolme eri osiota. Projekti koostuu Rob-Ex-tuotannosuunnitteluovelluksen integroinnista automaatioon eli tuotannosuunnittelujärjestelmän konfiguroimisesta sekä tiedonsiirron muodostamisesta Rob-Ex:n ja automaatiota ohjaavan valvomosovelluksen välille. Tämän lisäksi kokonaisprojektiin kuuluu automaatiolaboratoriossa sijaitsevan kappaletavaratuotantolinjan PLC-logiikan uudistaminen ja uudenlaisen valvomosovelluksen luominen tiedonsiirtoineen.

Viimeisenä elementtinä kokonaisprojektiin sisältyy laitteiston 3D-mallinnus ja liittäminen fyysistä laitteistoa ohjaavaan logiikkasovellukseen, mikä mahdollistaa tuotannon simuloimisen virtuaalisessa ympäristössä fyysisen laitteiston käyttämisen sijaan. Projektissa käytetään erilaisia tiedonsiirtomenetelmiä yhteyksien muodostamiseksi eri järjestelmien välille. Projektin hierarkia rakentuu ISA95-mallissa määriteltyjen tasojen mukaisesti, mutta tason neljä ERP-järjestelmää simuloidaan tässä projektissa vain kevyesti. Projektin fyysinen kokonaisrakenne on määritelty tarkemmin kuvassa 1.



Kuva 1. Kaavio kokonaisprojektista.

## 2.2 Kuvaus omasta osuudesta

Tämän opinnäytetyön puitteissa keskitytään projektissa ISA-95-mallin tasolle kolme eli MES-tasolle sijoittuvan tuotannosuunnittelujärjestelmän integroimiseen automaatiolaitteiston kanssa. Käytännössä tämä siis tarkoittaa Novotekin Rob-Ex-tuotannosuunnittelujärjestelmään perehtymistä ja sen konfiguroimista vastaamaan automaatiolaboratoriossa käytettävää laitteistoa eli laitteiston ja siinä tuotettavien kapaleiden mallintamista ohjelmistoon valmiiksi mallikonfiguraatioiksi. Lisäksi integraatioon kuuluu tiedonsiirron toteuttaminen valvomon ja tuotannosuunnittelujärjestelmän

välillä, jonka avulla järjestelmässä suunnitellut tilaukset voidaan siirtää automaatiota ohjaavaan valvomoon ja tilatiedot tuotettavista tilauksista takaisinpäin Rob-Ex:lle halutulla aikaviiveellä. Tällöin tilauksia pystytään suunnittelemaan joustavasti ja tilausten valmistamisesta saatavien tietojen avulla aikatauluttamaan tuotantoa entistä tehokkaammin.

Tiedonsiirron toteuttamiseen kuuluu myös erilaisten mahdollisten tiedonsiirtotekniikoiden tutkiminen teoriassa ja parhaiten käyttötarkoitukseen sopivan menetelmän valitseminen. Valittua tiedonsiirtomenetelmää koskien täytyy myös luoda Microsoftin Visual Studio -sovelluskehitysympäristöllä Windows-sovellus, joka toimii rajapintana automaatiota ohjaavan valvomon ja Rob-Ex-tuotannonohjausjärjestelmän välillä. Muita projektiin kuuluvia tehtäviä ovat Rob-Ex:n käyttämiseen perehdyttävien laboratorioharjoitusten luominen myöhempään opetuskäyttöön sekä erilaiset ohjeet ja muu dokumentaatio järjestelmän käyttämistä varten. Lisäksi tehtäväkuvaan kuului simuloida ERP-järjestelmältä saatavia tilauksia kevyesti.

### 2.3 Projektin päämäärä

Kokonaisprojektin päämääränä on saada luotua Metropolia Ammattikorkeakoulun automaatiotekniikan koulutusohjelman käyttöön järjestelmä, jonka avulla voidaan simuloida tuotannonohjausta sekä itse tuotantoprosessia ja sitä kautta kouluttaa alalle lisää osaajia. Projektin tavoitteena on myös osittain päivittää koulutusohjelman käyttämää laitteistoa ja ohjelmistoja sekä laajentaa erilaisten automaatiolaboratoriossa käytössä olevien järjestelmien spektriä entisestään.

Tämän osion päämääränä kokonaisprojektista on tehdä selvitystä tuotannonohjauksen periaatteista ja integrointitavoista automaatioon. Käytännön päämääränä on luoda malli käytettävästä laitteistosta Rob-Ex-tuotannosuunnittelujärjestelmään ja toteuttaa toimiva integraatio eli tiedonsiirto Rob-Ex:n ja automaation välillä. Tiedonsiirron toteuttamiseksi tulee selvittää käyttöympäristöön parhaiten soveltuva tiedonsiirtomenetelmä sekä toteutustapa. Lisäksi integraatiossa tulee saavuttaa sujuva toiminta, oikeiden parametrien välittäminen ja mahdollisten datahäviöiden eliminointi.

Projektin muihin tavoitteisiin kuuluvat lisäksi Rob-Ex-ohjelmiston toimintaan ja käyttämiseen liittyvien ominaisuuksien selvittäminen sekä ohjelmiston soveltamiseen liittyvän

tietotaidon kartuttaminen ja mahdollisten laajennusten kartoittaminen tulevaisuuden kannalta.

#### 2.4 Projektin toimintasuunnitelma

Projekti aloitetaan ensin kuhunkin käytettävään ohjelmistoon sekä laitteistoon perehtymällä, minkä jälkeen kartoitetaan projektin työvaiheita, arvioidaan projektin aikataulusta ja luodaan sen perusteella suunnitelma toteutuksen vaiheista. Suunnitelmaan kirjataan kuvaus projektista, sen tavoitteet, alustava projektirakenne sekä projektin aikataulu, jota yritetään noudattaa mahdollisimman hyvin työn sallimissa puitteissa.

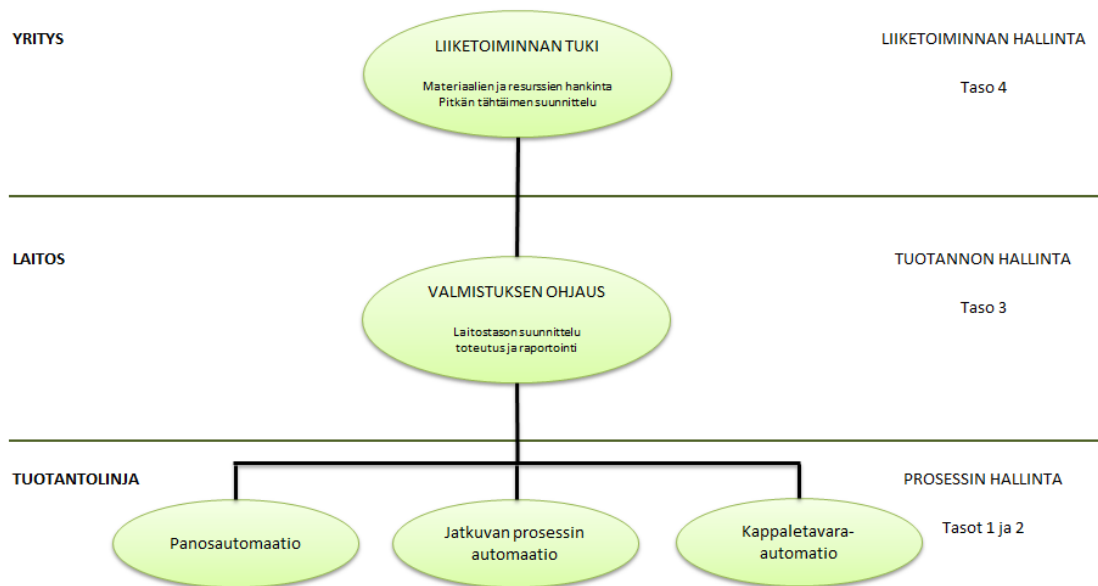
Muuta suunnittelua työhön liittyen tapahtuu projektiin osallistuvien osapuolten kesken. Tämä on oleellista, koska kyseessä on kaiken kaikkiaan yhteensä kolmesta eri lopputyöstä koostuva kokonaistoteutus, jonka eri osien tulee toimia keskenään. Yhteissuunnitteluun sisältyy esimerkiksi alustavaa pohdintaa siitä, missä vaiheissa ja miten yhteistyötä tehdään. Suunnitelmien ja aikataulun ollessa valmiina, projektia lähdetään toteuttamaan tutkimalla ja kehittämällä ratkaisua, joka täyttää projektin asettamat vaatimukset ja tavoitteet. Dokumentointia projektin etenemisestä suoritetaan säännöllisesti aina työn edistyessä, jotta projektista on helpompi muodostaa kokonaiskuva.

### 3 Käsitteistä

#### 3.1 ISA-95-malli

##### 3.1.1 ISA-95-standardi

ISA-95-malli on v.1945 perustetun ISA:n (International Society of Automation) kehittämä standardi, joka määrittelee yrityksen sisäisten hallinnollisten ja tuotannollisten osien, sekä kolmansien osapuolien ja toisten yritysten välistä integraatiota. ISA-95:n päämääränä on tarjota yritysmaailmalle kaikin puolin edullinen tapa eri sovellutusten integroimiseksi keskenään. Tavoitteena on luoda tarvittavan tekniikan tarjoamisen lisäksi yleinen informaatiomalli, jonka perusteella tietoa vaihdetaan eri osapuolten välillä. Standardin ideana on toimia lähtökohtana, jonka perusteella kullekin yritykselle luotava yksilöllinen toimintamalli on hyvä rakentaa. Standardi keskittyy välitettävän tiedon sisältöön ja jakaa tiedonvälityksen hierarkiaksi eri tasoille kuvan 2 mukaisesti. [2;7]



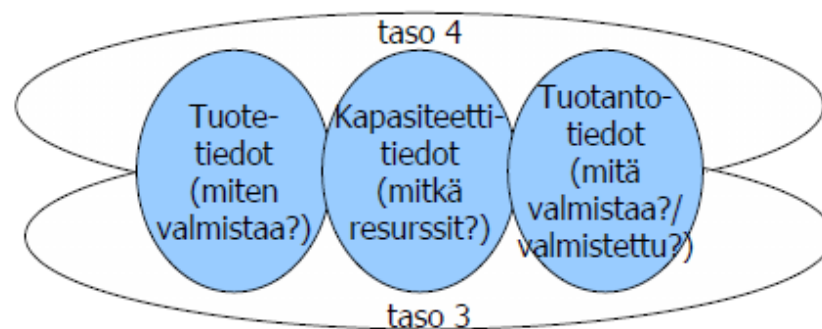
Kuva 2. ISA-95. Tuotannon toiminnallinen hierarkiamalli. [13]

ISA-95 on keskeinen informaatiomalli tuotannon automaatio- ja tietojärjestelmien integroinnin kannalta, jota tarvitsevat niin yritykset, laitetoimittajat kuin myös järjestelmien suunnittelijat. Malli ei ota niinkään kantaa käytettävään tiedonsiirtotekniikkaan, mutta suosittelee tiettyjä tekniikoita käytettäväksi. [2;13]



### 3.1.2 ISA-95:n -rakenne

ISA-95-standardi rakentuu viidestä osasta, joiden avulla yrityksen tietojen välitystä koskevat rajat voidaan hahmottaa sekä eri tasojen osapuolten ja niiden kommunikointiin tarvittavan tiedonsiirron rakenne ja sisältö määrittää. Standardi keskittyy pääosin B2M-tiedonvälityksen määrittämiseen tasojen 4 ja 3 eli yrityksen hallinnollisen ja tuotannollisen tason välille. Standardi määrittää tarvittavat tietovirrat eli mitä tietoa ja miten näiden tasojen välillä ja sisällä on hyvä välittää kuvan 3 mukaisesti, jotta tuotannon joustavuus ja tehokkuus saadaan optimoitua. [2]



Kuva 3. ISA-95-standardin määrittelemä geneerinen tietojen vaihdon sisältö [12;13]

Standardi koostuu seuraavista osioista:

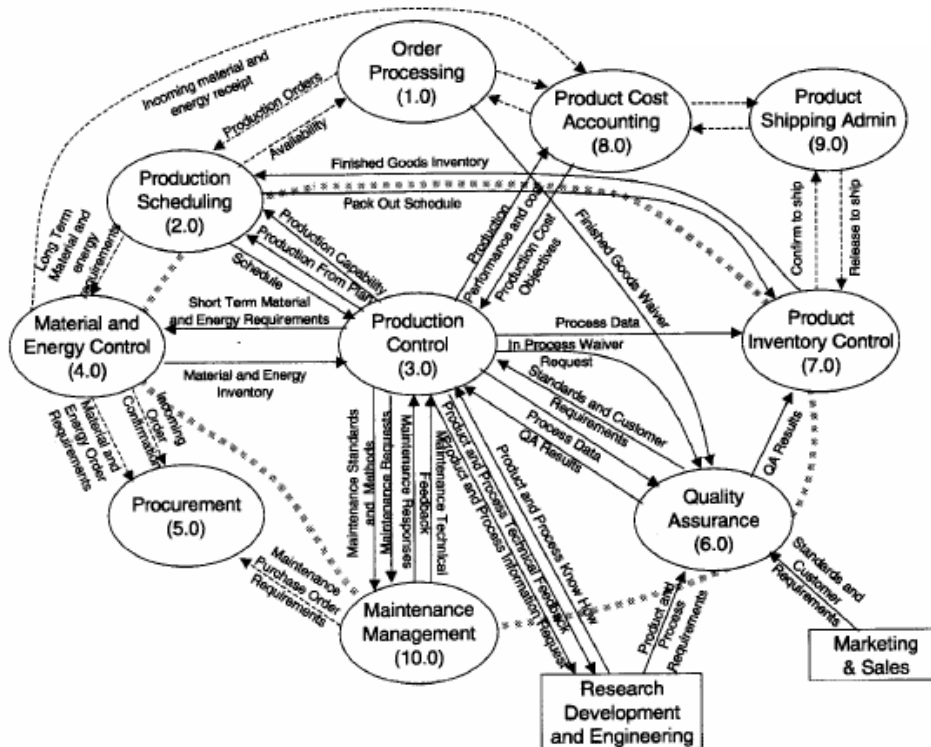
- ANSI/ISA-95.00.01-2010 Enterprise-Control System Integration – Osa 1: Models and Terminology sisältää standardikäsitteet ja tietomallit, joita voidaan käyttää määriteltäessä, mitä tietoa haluttuihin tuotantotoimintoihin tarvitaan ja mitä tietoa siirretään järjestelmien ja niiden osien välillä.
- ANSI/ISA-95.00.02-2010 Enterprise-Control System Integration – Osa 2: Object Model Attributes koostuu tietomallien objektien attribuuttien määrittelyistä. Attribuuttimäärittelyjen avulla voidaan yksityiskohtaisesti suunnitella eri järjestelmien välillä siirrettävät tietosisällöt.
- ANSI/ISA-95.00.03-2005, Enterprise-Control System Integration, Osa 3: Models of Manufacturing Operations Management käsittää tason 3 (MES-taso) toimintojen määrittelyyn. Sitä voidaan käyttää hyväksi kuvattaessa yleisesti MES-tason toimintoja ja vertailtaessa eri järjestelmätoteutuksia.
- Enterprise-Control System Integration - Osa 4: Object models and attributes of manufacturing operations management tarkoittaa osaa 3 ja määrittelee, mitä tietoa siirretään MES-järjestelmän eri osien välillä.

- ANSI/ISA-95.00.05-2007 - Enterprise-Control System Integration Osa 5: Business-to-Manufacturing Transactions määrittelee tason 3 ja 4 välillä siirrettävien tietojen muodon ja välitystavan.

[8;12]

Standardi kuitenkin kehittyä jatkuvasti, ja siihen luodaan uusia osia täydentämään mallia yhä monipuolisemmaksi ja kattavammaksi sekä paremmin soveltuvaksi vastaamaan todellisuutta. [2]

ISA95-standardin tarjoama malli jaottelee yrityksen tuotannon kahteentoista eri osakokonaisuuteen ja määrittelee eri osien välillä siirrettävät tiedot kuvan 4 mukaisesti. [2]



Kuva 4. ISA-95-standardin määrittelemät tuotannon osat ja tietovirrat osien välillä. [12]

Kaaviossa keskellä katkoviivan sisällä oleva alue kuvaa MES-tason eli valmistuksenohjauksen (taso 3) toimintoja ja ulompi alue katkoviivan ulkopuolella ERP-tason eli toiminnanohjauksen (taso 4) toimintoja. Kaavioon sijoitetut nuolet ja niiden yhteyteen liitetty selvennykset kertovat, mitä tietoja ja mihin suuntaan olisi mallin mukaan eri osien välillä hyvä siirtää. [2]

## 3.2 ERP-järjestelmät

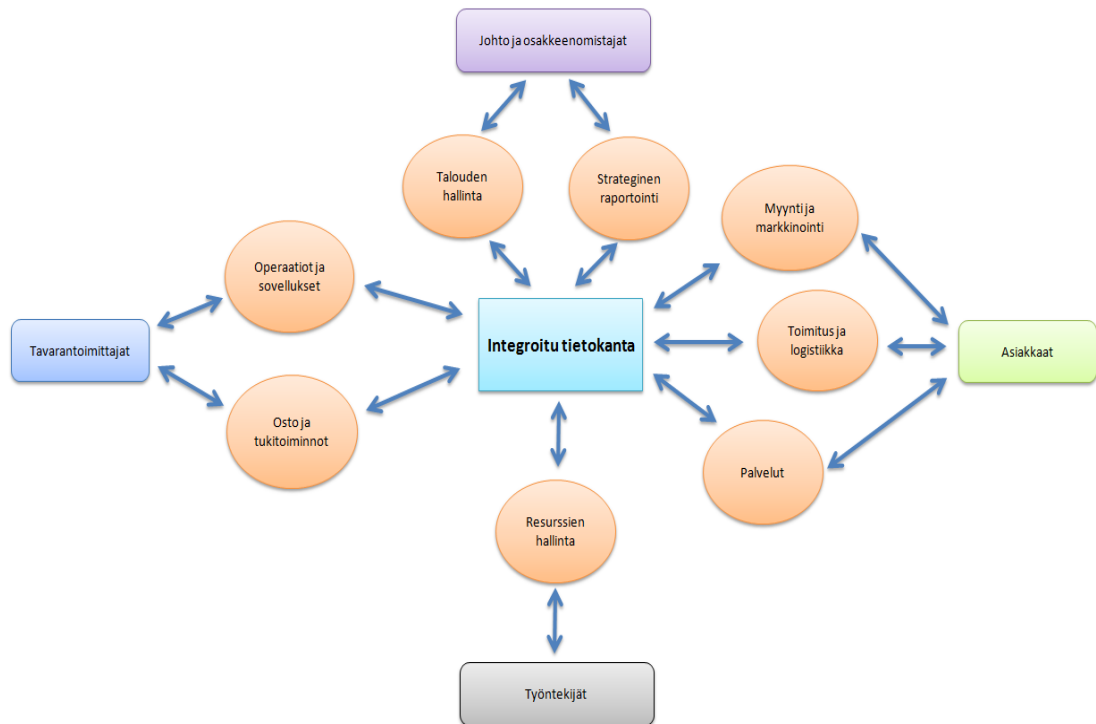
### 3.2.1 ERP

ERP:llä (Enterprise Resource Planning) eli toiminnanohjausjärjestelmällä tarkoitetaan yrityksen ydintoimintoihin liittyvän tiedon käsittelyyn tarkoitettua tietojärjestelmää. Toiminnanohjausjärjestelmän ideana on integroida kaikki yrityksen toimintaan liittyvä tieto yhdeksi hallittavaksi kokonaisuudeksi. ERP-järjestelmän konsepti kehittyi, kun erilaisia toiminnanohjaukseen liittyviä laskelmia alettiin suorittaa tietokoneavusteisesti. Myöhemmin tietoverkkojen ja erityisesti lähiverkon (LAN) kehitys mahdollisti tietojen välittämisen yrityksen eri osien välillä ja mahdollisti täten sisäisen integraation osien toiminnanohjauksessa. [19]

### 3.2.2 Toimintaperiaate

Toiminnanohjauksessa olennaisinta on sen itsensä sekä yrityksen liiketoiminnan tuottamien suurien tietomäärien hallinta. On tärkeää, että kaikki kerätty tieto tuodaan yhteen, jotta saadaan aikaiseksi kuva siitä, milloin, missä, kenen toimesta ja millä resursseilla jotakin pitäisi yrityksessä tehdä. Tämän selvittäminen on ERP-järjestelmän keskeinen toimintaperiaate. [19]

ERP-järjestelmät integroivat yritysten tietokantoja client–server-periaatteella laajamittaisesti, jotta yhdessä yrityksen osassa tehty päätös näkyisi muissakin yrityksen osissa. Lisäksi järjestelmät voivat olla yhteydessä yrityksen ulkopuolelle esimerkiksi tarjoamalla tietoa asiakkaille tai keräämässä tietoa alihankkijoiden ja muiden tavarantoimittajien tietokannoista. ERP-järjestelmien tarkoituksena onkin yhdistää yrityksen eri toimintojen hallitseminen yhdeksi kokonaisuudeksi kuvan 5 mukaisesti, jotta keskenään riippuvaisien prosessien suorituskykyä voidaan parantaa ja näin ollen yrityksen toiminnan kannattavuutta kohentaa. [19]



Kuva 5. ERP-järjestelmän keräämiä ja välittämiä tietoja yrityksessä [19]

### 3.2.3 Etuja ja haittoja

ERP-järjestelmien tarjoamia etuja ovat esimerkiksi yrityksen sisäisten tietojen läpinäkyvyys ja sen tarjoama parempi kommunikaatio yrityksen eri osien välillä. Järjestelmät mahdollistavat myös yritysten toimintojen paremman hallinnan tarkentamalla yrityksessä suoritettavia toimenpiteitä ja yrityksen sisäistä tiedonvälitystä. Tarkempi optimoitu toimintamalli parantaa osaltaan yrityksen toiminnan kannattavuutta ja sitä kautta kilpailukykyä markkinoilla. [19]

Toiminnanohjausjärjestelmien varjopuolia ovat niiden vaatimat tekniset toteutukset ja soveltuvuus erilaisiin toimintaympäristöihin. Järjestelmien mukauttaminen yrityksen tarpeisiin ja integraatio mahdollisesti vanhoihin järjestelmiin on kallista ja vie paljon aikaa. Onkin mahdollista, että tietyissä olosuhteissa järjestelmään sijoitettu investointi ei maksa itseään ikinä takaisin. Pahimmassa tapauksessa järjestelmän mukauttamiseen kuluneena aikana ehtii järjestelmä vanhentua ja yritys kärsiä tappioita investoinnistaan. ERP-järjestelmää jo olemassa olevaan yritykseen asennettaessa tulisikin siis tehdä kattava selvitys siitä, kannattaako järjestelmän käyttöönotto kohteessa pitkällä tähtäimellä laisinkaan. [19]

### 3.3 MES-järjestelmät

#### 3.3.1 MES

MES (Manufacturing Execution System) -termillä tarkoitetaan tehtaan tuotannon operatiiviseen ohjaamiseen käytettäviä toimintoja. MES-järjestelmät tarkoittavat lähes reaaliaikaisesti tuotteiden valmistusta seuraavia ja sitä ohjaavia valmistuksenohjausjärjestelmiä. MES-järjestelmät sijoittuvat ISA-95-mallissa tasolle kolme, eli toiminnanohjaustason ja tehdastason automaation välille. Valmistuksenohjausjärjestelmät pohjautuvat 1980-luvulla kehitettyihin ja yrityksissä käytettyihin CIM (Computer Integrated Manufacturing) -tiedonkeruujärjestelmiin, joita sovellettiin tuotannon suunnitteluun sekä henkilöstö- ja yrityshallintoon. Järjestelmistä muodostui lopulta erilaisia yhdistelmiä joiden pohjalta nykyinen MES-konsepti sai alkunsa. [13;11]

#### 3.3.2 Rakenne & toimintaperiaate

Valmistuksenohjausjärjestelmien toiminta perustuu tietokantaan sekä server–client-toimintaperiaatteeseen. MES -järjestelmät ovat yhdestä tai useammasta osasta koostuvia ohjelmistopohjaisia tietojärjestelmiä. Ne sisältävät ohjaus- ja tiedonkeruutoimintoja sekä tiedonsiirto- ja liitäntäteknikat, joiden avulla kommunikointi ISA-95-hierarkiamallissa ylöspäin toiminnanohjaustason ERP-järjestelmiin sekä alaspäin tehdastason automaatioon mahdollistuu. Tiedonsiirron mahdollistamiseksi vaativat MES-järjestelmät yleensä toimiakseen yrityksen sisäisen lähiverkon. MES-järjestelmille on myös ominaista tietynasteinen mukautettavuus, jonka avulla järjestelmien soveltaminen erilaisissa yritysten tuotantomalleissa onnistuu. [13;11]

MES-järjestelmät käsittelevät ERP-järjestelmistä saatavaa tietoa ja ohjaavat automaatiojärjestelmiä sekä fyysistä tuotantoa. Järjestelmät keräävät tietoa automaatiolta lähes reaaliajassa ja ovatkin nopeammin reagoivia verrattuna ERP-järjestelmiin. Tämä on keskeistä, sillä lähestyttäessä tuotteiden toimitusajankohtaa kasvaa valmistuksen suhteen tehtävien päätösten frekvenssi olennaisesti. Prosessituotannossa laitteiden ja prosessin osien hallinta nousee keskeiseksi MES-järjestelmien toiminta-alueeksi, kun taas kappaletavaruotannossa järjestelmät toimivat enemmänkin tiedonvälitysroolissa. Järjestelmien haasteena on sopeutua toimimaan erilaisissa tuotantomuodoissa eli jatkuvassa eräluontoisessa sekä yksittäistuotannossa. [13;11]

### 3.3.3 Toiminnallisuudet

MES-järjestelmien toiminnallisuudet voidaan jakaa yleisesti kolmeen osaan, joita ovat tuotannon toiminnallisuudet, laadun toiminnallisuudet sekä henkilöstön/resurssien hallinnan toiminnallisuudet. Lisäksi määritellyt toiminnallisuudet voidaan jakaa edelleen järjestelmän ydin- sekä tukitoimintoihin.

MES -järjestelmän ydintoimintoja ovat seuraavat:

- Liittymä toiminnansuunnittelujärjestelmään (Planning System Interface)
- Tuotantotilauksenhallinta (Work Order Management)
- Työpisteiden hallinta (Work Station Management)
- Varastojen hallinta (Inventory Tracking and Management)
- Materiaalin kulun hallinta (Material Movement Management)
- Tiedonkeruu (Data Collection)
- Poikkeustilanteiden hallinta. (Exception Management)

MES -järjestelmän tukitoimintoja ovat seuraavat:

- Kunnossapito (Maintenance Management)
- Aikaseuranta (Time and Attendance)
- Tilastollinen laadunohjaus (Statistical Process Control)
- Laadun varmistus (Quality Assurance)
- Prosessin tiedon ja suorituskyvyn analysointi (Process Data / Performance Analysis)
- Dokumenttien ja tuotetiedon hallinta (Documentation / Product Data Management)
- Tuotteen jäljitettävyys (Genealogy / Product Traceability)
- Alihankkijaseuranta. (Supplier Management)

### 3.3.4 MES -järjestelmien tarjoamat edut

MES-järjestelmien tarjoamien toimintojen avulla yrityksen valmistusprosessia voidaan tehostaa lisäämällä joustavuutta sekä selkeyttä. Tämän seurauksena tuotannon kannattavuutta ja yrityksen kilpailukykyä on mahdollista kohottaa. MES-järjestelmien avulla yrityksen eri tasojen integraatio tiivistyy tietoja reaaliaikaisesti fyysisestä tuotannosta kerättyinä ja sinne välitettäessä. Tällöin tuotteiden laatua ja läpimenoaika voidaan parantaa sekä mahdolliset laatuvirheet jäljittää. Lisäksi valmistusprosessia pystytään kehittämään optimaalisemmaksi mm. MES-järjestelmän suorittaman tiedonkeruun mahdollistaman OEE-analyysin avulla. [13;11]

## 3.4 Automaatio

### 3.4.1 Määritelmä

Automaatiolla voidaan tarkoittaa useita erilaisia automaattisia toimintoja, joihin voi sisältyä mm. itsestään tapahtuvaa tai mikroprosessori- ja tietokoneohjattua toimintaa, jossa käytetään esimerkiksi automatisoituja koneita, laitteita tai tuotantolinjoja mittauksiin perustuvan säädön avulla. Lähtökohtana automaatiossa ovat mitta-antureiden avulla tehdyt järjestelmän suureiden mittaukset, joista kerätään laitteiston ohjaamisessa tarvittavaa mittaustietoa. Mittausten lisäksi automaatio koostuu aina kerättyihin mittaustietoihin perustuvista ohjauksista tai säädöistä. Mikäli jompikumpi elementti järjestelmästä puuttuu, on kyseenalaista määritellä järjestelmä automaatiojärjestelmäksi. [10]

### 3.4.2 Automaation lajit

Teollisuuslaitoksissa sovellettava teollisuusautomaatio voidaan jakaa prosessi- ja kappaletavara-automaatioon. Teollisuusautomaatio koostuu fyysisesti teollisuuslaitoksen tuotanto- tai muuhun prosessiin sijoitetuista mitta- ja toimilaitteista sekä tietokonepohjaisista automaatiojärjestelmistä ja ohjelmistoista. Prosessiautomaatio keskittyy erilaisen virtaavien aineiden hallitsemiseen, kun taas kappaletavara-automaatiossa käsitellään selvästi erotettavissa olevia kappaleita. Näiden lisäksi on olemassa lukemattomia erilaisia automaatiosovelluksia, joita hyödynnetään joko teollisuudessa tai teollisuuden ulkopuolella esimerkiksi tutkimustyössä. [10]

### 3.4.3 Automaation käyttö

Automaation hyödyntäminen teollisuudessa kasvaa jatkuvasti, koska automaatio on yrityksille keskeinen kilpailukyvyn kohottaja. Automaation lisääntyminen on vähentänyt työpaikkoja, mutta sen avulla voidaan kuitenkin ylläpitää muuten kannattamatonta tuotantotoimintaa ja sitä tukevia työpaikkoja, niiden muuten kadotessa korkeiden henkilökustannusten omaavilta alueilta kokonaan. [10]

Automaation avulla voidaan suorittaa tehtäviä, jotka olisivat ihmiselle liian vaikeita, epämiellyttäviä tai jopa mahdottomia. Esimerkiksi mikrosirujen valmistaminen käsin tai yksitoikkoinen ympärivuorokautinen työ olisivat todennäköisesti ongelmallisia jättää ihmisten tehtäviksi. Tällöin mikäli hankalan työtehtävän automatisointi on olosuhteisiin nähden kannattavasti toteutettavissa, ei ole olemassa järkevää syytä, miksi jättää työ ihmisen tehtäväksi. Automaation myötä työtehtävät muuttuvat myös mielekkäämmiksi, koska ihminen vapautuu epämiellyttävien tehtävien suorittamisesta ja voi keskittyä muihin monipuolisempiin ja mielenkiintoisempiin töihin. Automaatoratkaisut tulee kuitenkin toteuttaa ihmisten ehdoilla siten, että niiden käyttäminen on johdonmukaista ja turvallista. [10]



## 4 Erilaiset tiedonsiirto ja integrointimahdollisuudet

### 4.1 SQL-tietokannat

#### 4.1.1 SQL-perusta

SQL (Structured Query Language) on lähes kaikkien tietokantojen taustalla käytettävä ohjelmointikieli, joka yleistyy jatkuvasti. Kielen yleistymiseen vaikuttavat myös osaltaan ERP- ja muiden tietokantoja hyödyntävien toiminnan- sekä tuotannonohjausjärjestelmien yleistyminen yritysmaailmassa. Kaikki suuret alan yhtiöt, mm. Microsoft, IBM ja Oracle tukevat järjestelmissään SQL-kieltä. Ohjelmointikieli sai alkunsa IBM:n tutkijan E. F. Coddin v.1970 kehittämästä relaatiomallista, ja ANSI on sittemmin standardisoinut kielen. SQL-kielestä on kuitenkin olemassa useita eri osapuolien luomia ja toisistaan hieman poikkeavia versioita. [11]

#### 4.1.2 Toimintaperiaate ja käyttö

SQL-tietokantojen toiminta pohjautuu relaatiomalliin perustuviin taulukoituihin tietoihin. Taulukot ovat linkitettyinä toisiinsa perusavaimien perusteella, joiden avulla voidaan määrittää, mihin taulukon riviin jonkin toisen taulukon tiedot kuuluvat tai viittaavat. Koska tietokannat pohjautuvat relaatiomalliin, joka on luotu joukko-opin perusteella, voidaan tietoja hakea tai muokata suorittamalla niille erilaisia joukko-operaatioita. [4]

SQL-tietokantoja voidaan lukea ja muokata SQL-kielen avulla. Lukeminen tapahtuu siten, että tietokannasta haetaan tiedot suorittamalla tietokantakysely ja tietojen lisääminen päivittämällä tietokantaa. Tiedot voidaan hakea tai niitä voidaan päivittää kantaa käyttäjän tai sovelluksen toimesta. Nämä operaatiot voidaan suorittaa jaotteleamalla tietokannassa olevat tiedot niiden sisältämän tiedon perusteella eri joukkoihin ja suorittaa esimerkiksi kyselyitä, jotka hakevat vain tietyt kyselyyn määritetyt kriteerit täyttävät tiedot tietokannasta. [13]

### 4.2 ODBC ja SQL

ODBC (Open Database Connectivity) on Microsoftin kehittämä standardisoitu rajapinta, joka tukee useita tietokantoja Windows-ympäristössä. ODBC:n avulla esimerkiksi eri

SQL-kannat ja kielet voidaan muuntaa yleiseen muotoon, joka mahdollistaa eri tuotteiden yhteensopivuuden keskenään. ODBC kykenee lukemaan helposti tietokannan taulu/sarakekuvaukset, joten se soveltuu hyvin raportti/kyselykäyttöön. Lisäksi kyselyitä voidaan suorittaa lukuisista ohjelmointiympäristöistä, joka monipuolistaa ja helpottaa SQL-tietokantojen käyttöä. [4]

### 4.3 Webservice -tekniikka

Webservicet ovat alustariippumattomia sovellusten osia, joiden avulla tietoa voidaan välittää verkon ylitse avoimia tiedonsiirtoprotokollia käyttäen. Niiden ideana on mahdollistaa erilaisten sovellusten käyttäminen verkon välityksellä. Webservicet perustuvat XML-formaatissa olevan tiedon välittämiseen HTTP (Hypertext Transfer Protocol) -protokollan ylitse. Webservice pohjautuu kolmeen eri tekniikkaan. Tiedon välittämiseen verkon yli käytetään SOAP (Simple Access Object Protocol) -protokollaa, joka on XML-pohjainen Internetin yli XML-muodossa välitettävää tietoa varten suunniteltu ja W3C:n standardoima protokolla. Kaksi muuta Webservicen yhteydessä käytettävää tekniikkaa ovat W3C:n standardoima Webservicen kuvailuun ja paikallistamiseen käytettävä WSDL (Web Services Description Language) sekä Microsoftin UDDI (Universal Description, Discovery and Integration), joka on yleinen Webservice-hakemisto. [23]

### 4.4 OPC-tekniikka

#### 4.4.1 OPC-standardit

OPC on eri automaatioalan yritysten ja Microsoftin yhteistyössä teollisuuden tarpeisiin v. 1996 kehittämä tiedonsiirtotekniikoiden standardisoitu kokoelma, jota ylläpitää samoista jäsenyrityksistä koostuva OPC Foundation -järjestö. OPC-lyhenne muodostuu sanoista Open Connectivity via Open Standards. Tekniikkaa käytetään teollisuudessa mm. kenttälaitteiden, erilaisten ohjausjärjestelmien sekä valvomoiden väliseen kommunikointiin. OPC koostuu erilaisista rajapintaa määrittelevistä standardeista, jotka määrittelevät tekniikan käyttämisen useimmissa teollisuuden vaatimissa sovelluksissa. [9;18]

Erilaisia OPC -rajapintaa määritteleviä standardeja ovat:

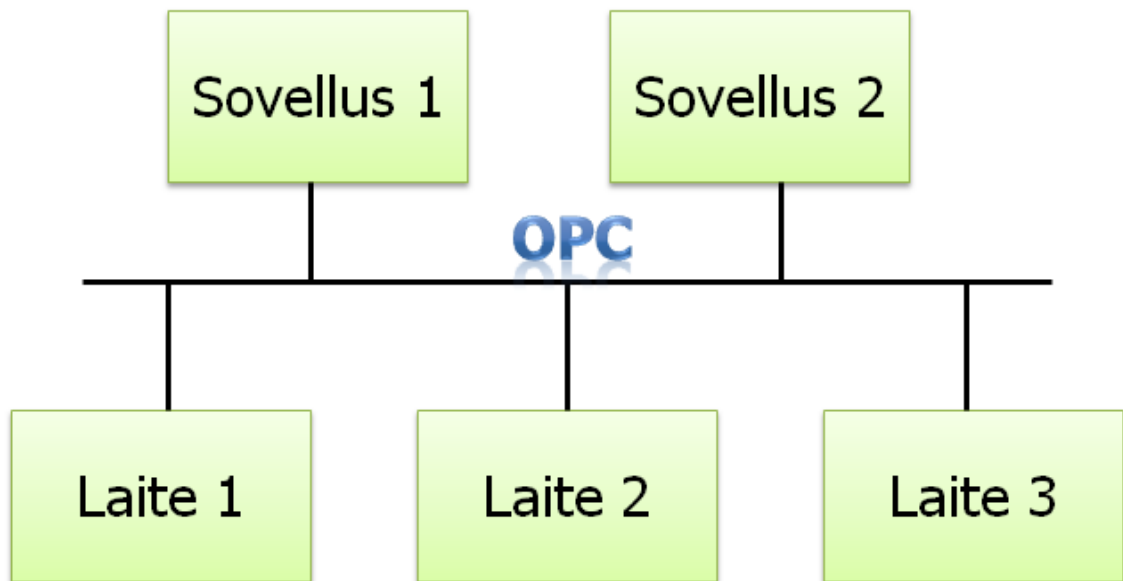
- OPC Data Access (OPC DA), jota käytetään reaaliaikaiseen tiedonvälitykseen teollisuuden laitteistojen ja käyttöliittymien välillä
- OPC Alarms & Events, joka määrittää hälytystietojen ja viestien välittämisen
- OPC Batch, jota käytetään eräluontoisesta tuotannosta koostuvassa teollisuudessa
- OPC Data eXchange, joka määrittää serverien välisen tiedonsiirron verkon välityksellä ja mahdollistaa diagnostiikan ja tilan valvomisen etäyhteydellä
- OPC Historical Data Access, joka tarjoaa historiatietojen hakemisen tallennetusta historiatietokannasta
- OPC Security, joka määrittää, kuinka suojata OPC serverin välittämiä tietoja
- OPC XML-DA, joka määrittää, kuinka tietoa voidaan esittää XML-muotoisena ja mahdollistaa myös tiedon siirtämisen verkon ylitse
- OPC Complex Data, joka määrittää, kuinka monimutkaisempia tietorakenteita, kuten binääri- ja XML-muotoista tietoa voidaan käyttää OPC-serverillä
- OPC Commands, kehitteillä oleva standardi, joka määrittää, miten laitteille voidaan lähettää ohjaukskomentoja
- OPC Unified Architecture (OPC UA), uusi standardimäärittely ja teknologia, jonka avulla OPC-rajapintaa voidaan käyttää alustasta riippumatta.

[9;18]

#### 4.4.2 OPC-toimintaperiaate

Teknologian toiminta perustuu server–client-periaatteeseen. OPC-serveri on tietokone-sovellus, joka muuttaa esimerkiksi kenttälaitteelta saadun datan OPC -protokollan mukaiseen standardisoituun muotoon. OPC-client on jokin sovellus, kuten valvomo, jolla on tarve lukea tietoa tai lähettää komentoja kenttälaitteelle. Client-sovellus lukee standardimuodossa olevat tiedot OPC-serveriltä ja voi lähettää tietoja takaisinpäin samassa muodossa, jotka OPC-serveri jälleen muuntaa kenttälaitteen ymmärtämään muotoon. Teknologia toimii siis tavallaan tulkkina erilaisten laitteistojen välillä, joskin OPC-

servereiden tulee tukea laitekohtaisia tietomuotoja. Tällöin laitteet voivat olla liitettynä järjestelmään väylämäisesti kuvan 6 mukaisesti. [16]



Kuva 6. OPC-tekniikan toimintaperiaate [13]

Vanhempi OPC Data Access-teknologia pohjautuu Microsoftin DCOM-teknologiaan, joten käyttöympäristöksi rajoittuu näin ollen Windows-ympäristö. OPC Unified Architecture taas on uusi seuraavan sukupolven tiedonsiirtoympäristö, joka on luotu vapauttamaan rajapinta useamman alustan käytettäväksi, tukien monia erilaisia tiedonvälitysteknologioita. [17]

#### 4.4.3 OPC-tarkoitus

OPC-tekniikan perimmäisenä tarkoituksena on tarjota sovellusten välille avoin liitettävyys siten, että erilaiset ja eri valmistajien tekemät sovellukset ja laitteet voivat vaihtaa keskenään tietoa käyttämällä yhteistä standardoitua rajapintaa. Tämän ansiosta jokaisesta sovellusta ja laitetta varten ei tarvitse erikseen luoda keskinäisiä ajureita. Standardisoidut OPC-rajapinnat kattavat yleisimmät teollisuuden tietojenvälityksen vaatimukset. [9]

OPC-tekniikka on tarkoitettu hyödynnettäväksi teollisuudessa esimerkiksi valvomosovelluksen ja kenttälaitteita ohjaavan logiikan tai valvomosovellusten keskinäisen tie-

donsiirron toteuttamiseen. Lisäksi tekniikkaa voidaan myös soveltaa MES- ja ERP-järjestelmien kytkemiseksi valvomosovelluksiin tai suoraan logiikoihin. [13]

## 4.5 XML

### 4.5.1 XML-standardi

XML on standardisoitu tiedostomuotoinen tiedon tallennus/välitystapa, jossa dokumentin sisältämän tiedon kuvaus voidaan liittää tiedon yhteyteen. Se määrittää XML-muotoisten tiedostojen rakenteen eli sen, millä periaatteella tietoa tulee kyseiseen tiedostoformaattiin tallentaa. Tunnus XML tulee sanoista eXtensible Markup Language ja sen kehitti W3C (World Wide Web Consortium) -järjestö v. 1996 aikaisemman SGML (Standard Generalized Markup Language) -kielen pohjalta. Sama järjestö myös ylläpitää standardia ja pyrkii lisäksi kehittämään muita uusia avoimia standardeja, joiden avulla Internetin ja verkottumisen kasvu ja kehitys voidaan taata pitkällä tähtäimellä. [21;22]

XML-standardin tavoitteita on toimia suoraviivaisesti Internetin välityksellä sekä tukea laajasti eri sovelluksia. Tämä saavutetaan pitämällä XML-muotoisina luotavat tiedostot mahdollisimman selkeinä ja suoraviivaisina standardia noudattaen ilman erillisiä lisäominaisuuksia. Tuloksena tiedostoja käyttävien ohjelmien luominen helpottuu ja tiedostojen selkeys sekä luettavuus säilyvät hyvinä. [21]

### 4.5.2 XML-muotoiset dokumentit

XML itsessään määrittää XML-muotoon luotavien dokumenttien tietorakenteen. Dokumenttien sanotaan olevan oikein muodostuneita, mikäli ne noudattavat yleistä XML-kirjoitusmuotoa. Dokumenttien kielioppi eli tietosisällölle ennalta määritelty muoto ja looginen järjestys voidaan tarkistaa vertaamalla XML-dokumenttia joko DTD (Document Type Definition) -tiedostoon tai tarkemmin tietojen oikeanlaisen ja loogisen rakenteen määrittelevään XML-skeemaan (XML schema). Mikäli tarkistettavat xml-muotoon kirjoitetut tiedostot ovat oikein muodostuneita ja niiden sisältämät tiedot täyttävät lisäksi oikeat vaatimukset, kutsutaan tiedostoja valideiksi. [3]

XML-muotoon kirjoitetut tiedostot muodostuvat elementeistä, elementtien sisälle kirjoitetusta tiedosta ja elementtien yhteyteen määritettävistä attribuuteista. Jokaisessa dokumentissa täytyy olla juurielementti, joka voi esiintyä dokumentissa vain kerran ja jonka sisällä kaikki muut elementit sijaitsevat. Myös juurielementtien sisällä olevien elementtien sisään voidaan sijoittaa lisää elementtejä ja muodostaa näin erilaisia tietorakenteita. Tiedostot rakentuvat käytännössä kuvan 7 mukaisesti.[3]

```
<?xml version="1.0"?>
- <Juurielementti>
  <Elementti Attribuutti="Attribuutin arvo"> Tietoa elementin sisällä. </Elementti>
</Juurielementti>
```

Kuva 7. XML-muotoinen esimerkkitiedosto.

Esimerkkitiedostossa määritetään ensin käytettävä XML-versio, seuraavalla rivillä on tiedoston juurielementti, jonka sisään on sijoitettu tavallinen tietoa sisältävä elementti, jota on kuvailtu attribuutin avulla. Elementit tulee aina sulkea kauttaviivalla merkityllä sulkemistagilla.

#### 4.6 CSV-tiedostomuoto

CSV (Comma Separated Values) -formaatti on yleinen tekstimuotoisena esitettävä tiedostomuoto, jonka sisältämät tiedot on jaoteltu peräkkäin ja eroteltu pilkuilla. Tiedostomuoto on helppokäyttöinen ja selkeä sekä helposti prosessoitavissa monien erilaisten sovellusten, kuten tietokantojen, kanssa. Tiedot voidaan lukea ja kirjoittaa tiedostosta tai tiedostoon esimerkiksi taulukkomuotoisina. Tiedostoformaatin käyttötapa ja muoto ovat käytännössä vapaasti määriteltävissä tarvetta vastaavaksi. Virallisen tiedostomuodon on kuitenkin määritellyt v. 2005 Internet-standardina IETF (Internet Engineering Task Force). Standardi RFC 4180 (Request For Comments: 4180) kuvailee CSV-tiedostomuodon jäsentelyä ja sen ominaisuuksia. RFC 4180 on myös Internet-protokollia ja osoitteita koordinoivan IANA:n (Internet Assigned Numbers Authority) vahvistama määritelmä. Koska CSV-tiedostot esitetään tekstimuodossa, täytyy huomioida niiden merkkijonojen oikeanlainen koodaus. Tiedostot rakentuvat käytännössä kuvan 8 mukaisesti. [20;6;5]

```
0128,12-02-2013 11:00,Metropolia,Kuutio metalli,225.0  
25-04-2013 15:00,syöttö,Syöttöasema,25-04-2013 15:05,Prässäys,Prässi,18-04-2013 08:13,Varastoon siirto,Varasto  
0256,18-03-2013 15:00,Kuutiotukku,Kuutio muovi,552.0  
26-04-2013 15:00,syöttö,Syöttöasema,25-04-2013 15:05,Prässäys,Prässi,18-04-2013 08:13,Varastoon siirto,Varasto
```

Kuva 8. CSV-muotoinen esimerkkiedosto.

## 5 Rob-Ex

### 5.1 Ominaisuudet

Rob-Ex-tuotannonsuunnittelujärjestelmää voidaan hyödyntää monenlaisissa sovelluksissa, joissa tarvitaan joustavaa ja helppokäyttöistä työkalua tuotannon tai muun toiminnan tarkkaan aikatauluttamiseen. Ohjelmiston ideana on toimia tehokkaana ajanhallintatyökaluna tuotantoa suunniteltaessa. Sen avulla voidaan aikatauluttaa tuotantoa havainnollisesti ja joustavasti sekä kokeilla erilaisia ratkaisuja siten, että optimaalinen tuotantosuunnitelma voidaan saavuttaa. Ohjelmisto sisältää myös joukon valmiita toimintoja, joiden avulla tuotannon suunnittelu nopeutuu ja siten tehostuu. [15]

Ohjelmiston avulla pystytään mallintamaan käytettävä tuotantolaitteisto eli resurssit, tuotantoajat kalenterimuotoisesti ja erilaiset tuotteet sekä niiden valmistamiseen tarvittavat operaatiot, reitit ja materiaalit. Näiden määritysten perusteella ohjelmiston toimintaa voidaan myös mukauttaa tarvittavalle tarkkuustasolle tuotantoympäristöstä riippuen. Rob-Ex voidaan lisäksi integroida esimerkiksi ERP -järjestelmään ja myös tarvittaessa automaatioon tai muihin järjestelmiin hyvien integrointimahdollisuuksiensa ansiosta. [15]

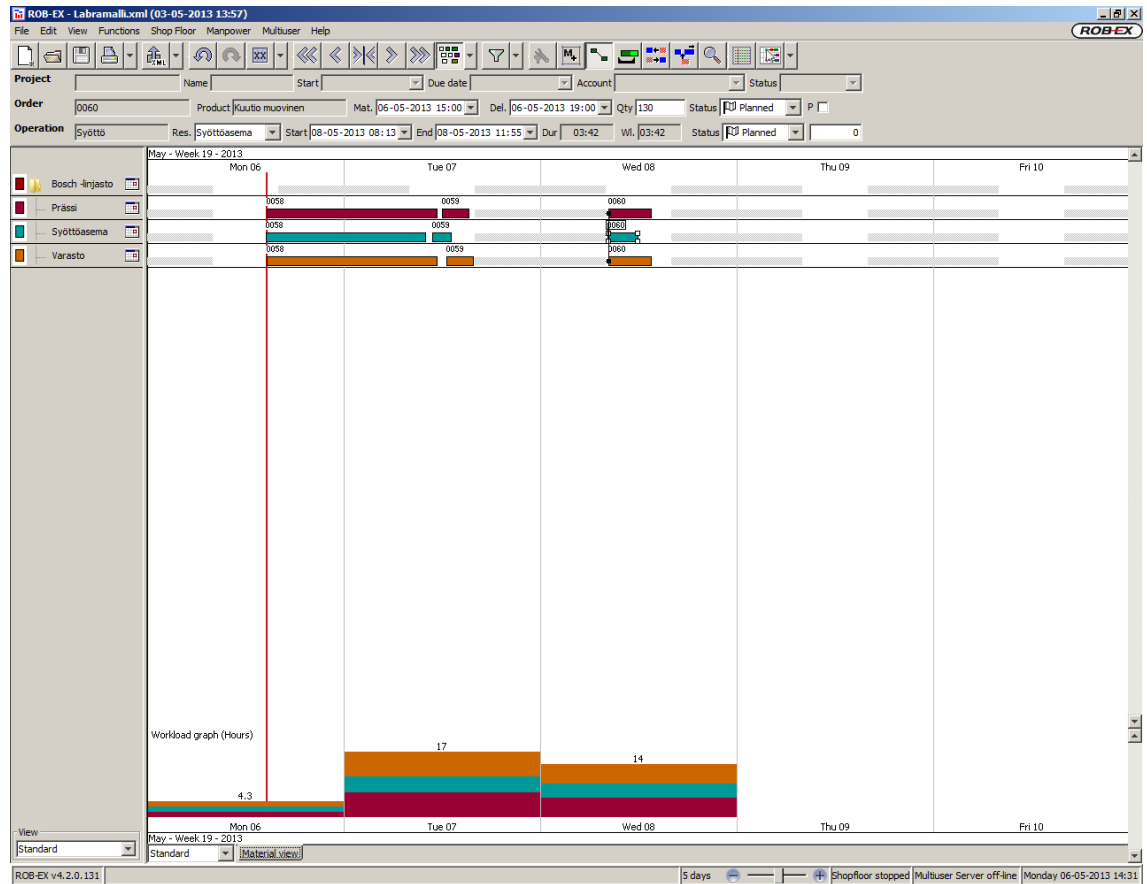
Ohjelmistoa käytetään server–client-periaatteella, jossa Rob-Ex on asennettuna serverikoneelle verkkosijaintiin, ja tuotannonsuunnittelua voidaan suorittaa eri tietokoneilta Rob-Ex Gantt Pro Planner -sovelluksilla. Suunnitelmia voidaan tarkkailla Rob-Ex Gantt Viewer -sovelluksella ja tilausten etenemistä päivittää tarvittaessa manuaalisesti selainpohjaisella Rob-Ex Gantt Shop Floor -sovelluksella. [15].

## 5.2 Toimintaperiaate

Rob-Ex käyttää tuotannosuunnittelussa hyväkseen Gantt-taulukkoa, jossa tilaukset näytetään graafisesti resurssikohtaisina erivärisinä palkkeina, jotka on sijoitettu aikajanalle kuvan 9 mukaisesti. Palkkien pituus aikajanalla ilmaisee, kuinka kauan kunkin resurssikohtaisen operaation ja sitä kautta tilauksen tuottaminen kestää. Itse tuotannosuunnittelu tapahtuu siten, että palkkeja siirrellään aikajanalla esimerkiksi hiiren avulla raahaamalla tai käyttämällä siirtämiseen valmiita toimintoja. Suunnittelu on tällä tavoin havainnollista sekä helppoa ja erilaisia suunnitelmaratkaisuja on vaivatonta keilla. [15]

Ohjelmisto laskee kunkin tilauksen kohdalla tuottamiseen tarvittavan ajan, materiaalit, sekä työkuorman ja sijoittaa tilauksen sen jälkeen tuotettavaksi määriteltynä ajankohdaksi, ottaen samalla huomioon kalenteriin määritetyt työajat. Ohjelmiston laskentatapa perustuu sille määritettyihin resursseihin, operaatioihin, materiaaleihin, työaikakalenteriin sekä reitteihin koskeviin parametreihin. Näihin parametreihin on määritetty, mitä resursseja eli tuotantolaitteita tai työvoimaa on käytettävissä ja milloin sekä kuinka tehokkaasti resurssit tuottavat aikayksikköä kohden. Lisäksi on määritetty, mitä operaatioita resurssien on mahdollista suorittaa ja mitä materiaaleja operaatiot kuluttavat tai tuottavat. Tuotekohtaisesti määritettynä on vielä reitti eli operaatiot, jotka tulee suorittaa tuotetta valmistettaessa. [15]





Kuva 9. Rob-Ex-tuotannonsuunnittelujärjestelmän käyttöliittymä.

### 5.3 Sovelluskohteet

Rob-Ex-tuotannonsuunnittelujärjestelmää voidaan soveltaa laajalti prosessi- tai kappale-tavaratuotantona suoritettavan tuotannon suunnittelemiseen. Ohjelmistoa on myös käytetty esimerkiksi asuntoja vuokraavan yrityksen suunnittelupohjana asuntojen vuokraamisen aikataulutamisessa. Rob-Ex:n etuna ja monipuolisen sovellettavuuden pohjana ovat juuri joustavuus, muokattavuus ja liitettävyyys, jotka saavat ohjelmiston soveltumaan monenlaisiin eri käyttötarkoituksiin. Ohjelmiston pääasiallinen käyttökohde muodostuu kuitenkin erilaisista tuotantolaitoksista, joissa tuotannon suunnitteleminen näyttää keskeistä roolia kannattavan toiminnan takaajana. [1]

#### 5.4 Integrointimahdollisuudet

Rob-Ex-tuotannosuunnittelujärjestelmä sisältää tuen monenlaisille eri integrointimenetelmille. Erilaisia integrointimahdollisuuksia ja tekniikoita ovat mm.

- XML-tiedostopohjainen tiedonsiirto
- CSV-tiedostopohjainen tiedonsiirto
- OPC-tiedonsiirto
- SQL-tietokantapohjainen tiedonsiirto
- Webservice-verkon yli tapahtuva tiedonsiirto.

Eri integrointimenetelmiä varten käytetään Rob-Ex:sta löytyviä Plug-in-toimintoja sekä ohjelmiston kanssa yhteensopivia serverisovelluksia. Rob-Ex voidaan integroida ERP-järjestelmän, tehdastason automaation sekä muiden tuotantoa tukevien järjestelmien kanssa. Tällöin saadaan joustava ja reaaliaikainen järjestelmä, jonka avulla tuotantoa voidaan optimoida tehokkaasti. [15]

## 6 Toteutus

### 6.1 Valittu integrointitekniikka

Integrointitekniikaksi Rob-Ex-tuotannosuunnittelujärjestelmän liittämiseksi automaatioon valittiin tässä projektissa tiedonsiirto tiedostopohjaisesti. Tekniikkaan päädyttiin sen selkeyden ja hyvien toteutusmahdollisuuksien sekä luotettavaksi ennakoitujen toiminnan perusteella. Etuina muihin integrointimenetelmiin verrattuna nähtiin valitussa tekniikassa mm. se, ettei tietokantoja tarvitse ylläpitää, että tietojen mahdollinen katoaminen on epätodennäköisempää ja että eri järjestelmiä voitaisiin simuloida tarvittaessa tiedostoja manipuloimalla. Kaiken kaikkiaan tekniikan koettiin soveltuvan hyvin käyttöympäristöönsä automaatiolaboratorion olosuhteisiin sekä opetuskäyttöön.

### 6.2 Toteutustapa

Integrointi toteutettiin käyttämällä XML- ja CSV-tiedostoformaatteja tietojen välittämisessä tuotannosuunnittelujärjestelmän ja automaation välillä. Tietojen välitystä ja oikeanlaista päivittämistä varten luotiin tuotannosuunnittelujärjestelmän ja automaatiota ohjaavan valvomosovelluksen väliin tiedonvälitysovellus, jonka avulla tietoja päivitettiin osapuolelta toiselle tiedostoja käsittelemällä.

Tarvittavat tiedostot saatiin luotua Rob-Ex-tuotannosuunnittelujärjestelmän valmiita makrotoimintoja hyväksikäyttäen sekä tiedonvälitysohjelman ja valvomosovelluksen kirjoittamina. Rob-Ex:n kirjoittamat ja lukemat tiedostot olivat XML-formaatissa ja valvomon lukemat sekä kirjoittamat tiedostot CSV-muodossa.

### 6.3 Tietojen välitys

Mahdollisia tuotannosuunnittelujärjestelmän ja automaatiolaitteistoa ohjaavan valvomosovelluksen välillä välitettäviä tietoja ovat:

Tilaustiedot Rob-Ex:lta valvomolle

- tilauksen ID-numero

- tilauksen tuottamisen aloitusajankohta
- asiakkaan nimi
- valmistettava kappalemäärä
- tuotenimike

#### Operaatiotiedot Rob-Ex:lta valvomolle

- operaation aloitusajankohta
- operaation nimi
- operaatiossa käytettävä resurssi

#### Tilautustiedot valvomolta Rob-Ex:lle

- tilauksen ID-numero
- tilauksen tilakoodi
- hukkatavaran määrä (tilauskohtainen)
- tuotettujen lukumäärä (resurssikohtainen)

#### Operaatiotiedot valvomolta Rob-Ex:lle

- operaation nimi
- operaatiossa käytettävä resurssi
- operaation tilakoodi
- tuotettujen lukumäärä
- hukkatavaran määrä

Kaikki luetellut tiedot on mahdollista välittää integrointia varten ohjelmoidun tiedonvälitysohjelman avulla. Työssä keskityttiin kuitenkin toistaiseksi vain tilauskohtaisten tietojen välitykseen niiden yksinkertaisuuden ja projektin kehitysluonteisuuden vuoksi. Toisin sanoen operatiokohtaisesti ei välitetty muita kuin valmistuneiden kappaleiden lukumäärää valvomolta Rob-Ex-tuotannosuunnittelujärjestelmälle. Kaikki muu välitetty tieto oli tilauskohtaista.

## 6.4 Toteutuksen eri vaiheet

### 6.4.1 Aloitus

Projektin toteutus alkoi helmikuun alussa 2013, jolloin projektityö vastaanotettiin ja sitä pohjustettiin ohjaavien opettajien kanssa. Työn toteutus itsessään alkoi ensin Rob-Ex -tuotannonsuunnitteluohjelmiston käyttöohjeen lukemisella sekä suunnittelulla tiedonsiirron toteuttamiseksi. Projektista luotiin myös alustava kirjallinen suunnitelma aikataulun sekä projektin tavoitteiden osalta. Tämän jälkeen oli vuorossa projektikoneen kasaminen automaatiolaboratorioon ja Rob-Ex-ohjelmistoon tutustuminen käytännössä.

### 6.4.2 Rob-Ex

Rob-Ex:ään tutustuminen käytännössä suoritettiin ohjelman eri toiminnallisuuksia kokeilemalla ja ohjelman käyttämisen opettelua varten suunniteltuja harjoitustehtäviä tekemällä. Ohjelman makrotoimintojen ominaisuuksia kokeiltiin kirjoittamalla tiedostoja ulos lukemalla tiedostoja jälleen Rob-Ex:ään. Tiedostojen parametreja muokkaamalla saatiin selville eri muuttujien vaikutusta esimerkiksi tilausten tilan suhteen Rob-Ex:n gantt -taulukkoon. Lisäksi tuotannonsuunnittelujärjestelmästä ulos kirjoitetuista tiedostoista selvisi tiedostomuoto, jota tulisi käyttää tiedonvälitystyökalun XML-tiedostojen lukemis- ja kirjoittamisominaisuuksia määriteltäessä.

### 6.4.3 Tiedonsiirtotyökalu

Tiedonsiirtotyökalun luominen aloitettiin luomalla tiedonsiirto Rob-Ex:n kanssa ja tekemällä ensin raakaversio sovelluksesta, jonka avulla perustoimintoja voitaisiin testata. Lähinnä raakaversioiden tavoitteena oli saada oikeat komennot ja syntaksit sekä käytettävä ohjelmistorakenne selville varsinaisen työkalun luomista varten.

Tiedonsiirtotyökalun kehittäminen venyi projektin aikana pitkälle kevääseen lisäominaisuuksien sekä odottamattomien virhetoimintojen korjaamisen ja lopullisen toimintamallin hiomisen vuoksi. Työkalua muutettaessa ja päivitetessä pidettiin versionumeroa sovelluksen yhteydessä vanhempien versioiden erottamiseksi uudemmista ja paremmista.

Työkalun ollessa valmiina luotiin sovellukselle vielä ohjeet, jotta käyttäminen onnistuu myös sovellukseen perehtymättömiltä osapuolilta. Ohjeiden kirjoituksen tuli sijoittua vasta tiedonsiirtotyökalun saadessa lopullisen muotonsa, ettei väärää tietoa pääse kulkemaan ohjeeseen.

#### 6.4.4 Raportointi

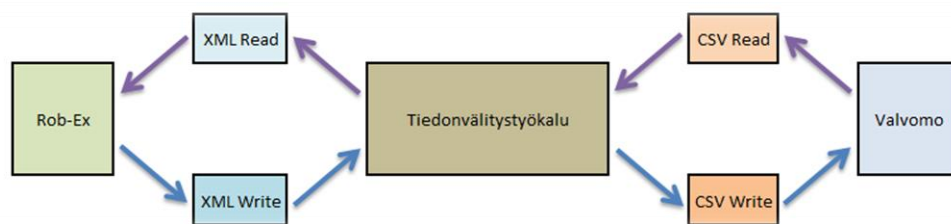
Projektin aikana kirjoitettiin muistiin tehdyt työt ja havaitut ongelmat lokitiedostoksi. Tällöin projektin tekijällä oli koko ajan kiintopiste siitä, missä vaiheessa projektia ollaan menossa ja mitä tulisi tehdä seuraavaksi. Lisäksi kaikki epäkohdat projektin aikana olivat listattuina, jotta niihin oli helppo perehtyä ja ongelmat ratkaista järjestelmällisesti.

Loppuraportti projektista luotiin pääosin vasta projektin loppuvaiheessa kerättyjen materiaalien perusteella. Tällöin projektista oli paljon materiaalia kirjoitettavaksi eikä itse projektin tekeminen häiriintynyt tai ristiriitaista tietoa päätyntä raporttiin.

## 7 Tiedonsiirtotyökalu

### 7.1 Työkalun toimintaperiaate

Tiedonsiirtotyökalu luotiin Microsoft Visual Studio 2010 Ultimate Edition -sovelluskehitysympäristöllä. Työkalun toimintaperiaatteena on tietojen lukeminen Rob-Ex-tuotannonohjausjärjestelmältä, tietojen välittäminen valvomolle, tuotantotietojen lukeminen valvomolta ja lopuksi tietojen päivittäminen takaisin Rob-Ex-ohjelmistoon kuvan 10 mukaisesti.



Kuva 10. Tiedonvälitystyökalun toimintaperiaate.

Yksinkertainen toimintaperiaate itsessään koostuu neljän eri tiedoston lukemisesta tai kirjoittamisesta. Rob-Ex Scheduler -ohjelmistolle on konfiguroitu valmis makrofunktio, joka tietyin aikavälein kirjoittaa sille määritettyyn kansioon XML-formaatissa tiedoston. Tästä tiedostosta työkalu lukee Rob-Ex:ään määritetyt tilaukset ja muut tarvittavat tiedot, tallentaa tiedot taulukoihin ja välittää tilaukset jälleen eteenpäin valvomolle tiedostona CSV-formaatissa. Seuraavaksi työkalu lukee valvomon kirjoittamat tuotantotiedot valvomolta saatavasta CSV-formaatissa olevasta tiedostosta ja päivittää tiedot jälleen sisäisiin taulukoihinsa. Viimeisenä työkalu kirjoittaa tiedot Rob-Ex-tuotannonohjausjärjestelmälle XML-formaatissa olevaan tiedostoon, jonka Rob-Ex:ään konfiguroitu toinen makrofunktio käy lukemassa.

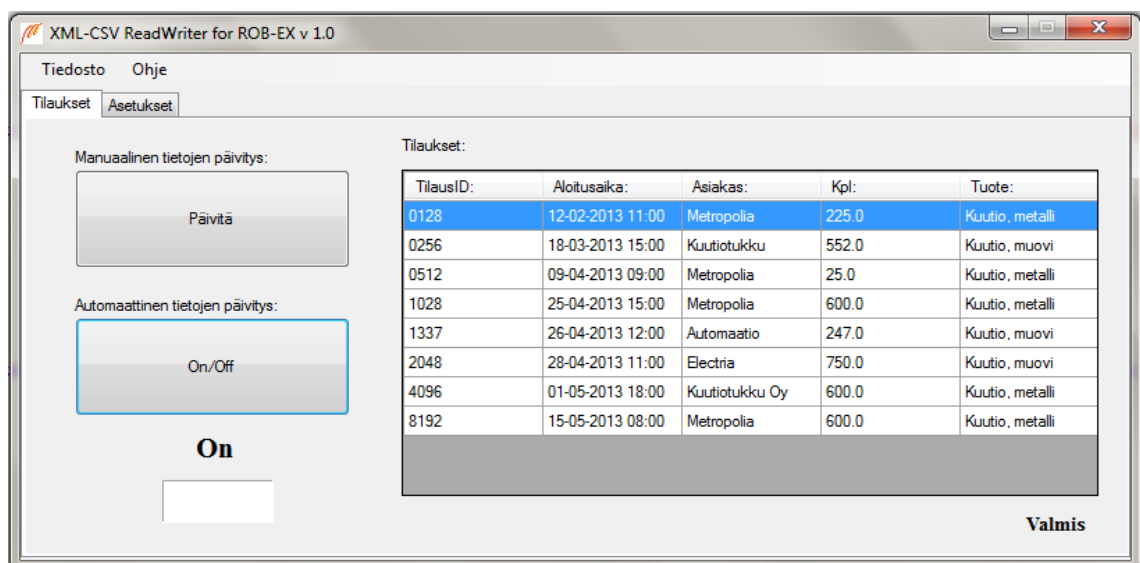
Tällä periaatteella on tiedostopohjainen tiedonsiirtorajapinta ja sitä kautta integraatio toteutettu Rob-Ex-ohjelmiston ja valvomon välillä joko automaattisesti tai manuaalipäivityksellä. Rob-Ex:n ja työkalusovelluksen välillä hyödynnetään automaattisessa tilassa oletuksena tiedoston muuttumista tunnistavaa ominaisuutta, jolloin työkalun päivitys-sykli ajetaan lävitse Rob-Ex:n kirjoittaman työkalulle luettavaksi tarkoitetun XML-

tiedoston päivittyessä. Tällöin työkalu päivittää syklin päätteeksi Rob-Ex:lle luettavaksi tarkoitetun XML-tiedoston valvomolta saatujen tuotantotietojen mukaisesti.

Vastaavasti Rob-Ex:ään päivitettyjä tietoja lukevaan valmiiseen makrofunktioon on myös määritetty tiedot luettavaksi silloin, kun luettava tiedosto muuttuu. Tällöin lopputuloksena saadaan automaattinen päivityssykli, joka pyörähtää aina tietoja tuotannon suunnittelujärjestelmästä tasaisin aikavälein valvomolle kirjoitettaessa. Tämä eliminoi teoriassa myös mahdolliset datahäviöt ja ohjelmistojen väliset konfliktit, joita pelkällä aikasyklillä ajettavat erilliset tiedon luku- ja kirjoitusfunktiot saattaisivat aiheuttaa. Manuaalipäivityksellä tietojen päivityssykli ajetaan läpi vain kertaalleen nappia painamalla.

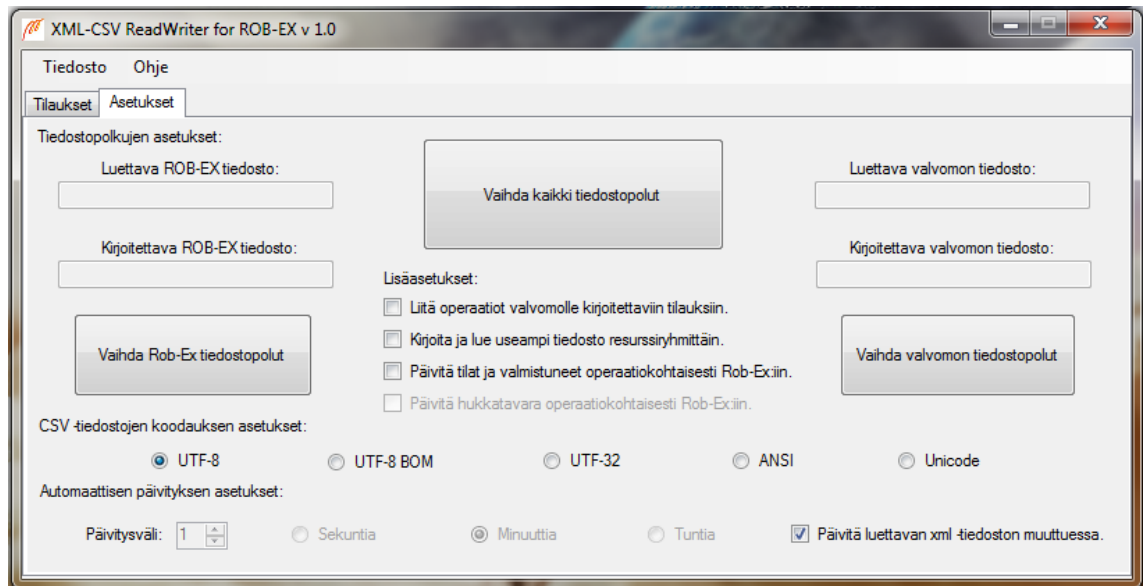
## 7.2 Ominaisuudet ja käyttö

Työkalun ominaisuuksiin kuuluvat automaattinen tai manuaalinen tietojen päivitys, tietojen päivityksen ajaminen automaattisesti joko aikasyklillä tai tiedoston muuttumista tunnistamalla. Lisäksi sovelluksesta löytyvät lisäasetukset, joiden avulla voidaan valvomolle kirjoitettavien tilausten yhteyteen liittää suoritettavat operaatiot, lukea ja kirjoittaa useampi valvomotiedosto Rob-Ex:ään määritettyjen resurssiryhmien eli eri tuotantopisteiden perusteella ja päivittää tietoja Rob-Ex:ään oletuksena olevan tilauspohjaisen päivityksen sijaan operaatiokohtaisesti. Ohjelman käyttö tapahtuu kuvissa 11 ja 12 esiintyvän käyttöliittymän avulla.



Kuva 11. Tiedonsiirtotyökalun käyttöliittymän tilaukset -välilehti.





Kuva 12. Tiedonsiirtotyökalun käyttöliittymän asetukset -välilehti.

Työkalulle määritetään tai luodaan aluksi kaikki tietojen välityksessä tarvittavat tiedostot. Rob-Ex-tuotannosuunnittelujärjestelmän ja työkalun välillä käytettäväksi tiedostoiksi määritetään Rob-Ex:n kirjoittama export- ja luettava import-tiedosto. Kommunikointitavasta riippuen, vaihdetaanko tietoja yhden vai useamman valvomon kanssa, määritetään jokaiselle valvomolle kirjoitettava ja valvomolta luettava tiedosto. Tämän jälkeen työkalu lukee ja kirjoittaa tietoja valvomon ja Rob-Ex:n välillä joko manuaalisesti päivittämällä ”Päivitä”-nappulaa painamalla tai automaattisesti ”On/Off”-nappulaa painamalla. Mikäli sovellus asetetaan automaattiajolle, toimii se oletuksena luettavaa Rob-Ex:n kirjoittamaa tiedostoa tarkkaillen ja tiedoston päivittyessä suorittaa päivityssyklin aina kertaalleen. Automaattiajon vaihtoehtona on asettaa päivityssyklille aikaväli 5 s - 24 h, jolloin päivityssykli ajetaan aina määritetyn aikavälin täytyessä. Valvomolle kirjoitettavien CSV-tiedostojen merkkien koodaus voidaan lisäksi valita yleisimpien standardien välillä.

## 7.3 Ohjelmointi

### 7.3.1 Ohjelmointikieli ja valmistelut

Tiedonvälitystyökalun ohjelmointiin valittiin käytettäväksi Microsoftin Visual Basic -ohjelmointikieli. Syitä tähän olivat ohjelmoijan aiempi kokemus kielen käytöstä opiskelunsa aikana sekä oletettavasti hyvä tuki erilaisten valmiiden ohjelmointikieleen sisälty-

vien toimintojen käyttämiseksi. Sovelluksen ohjelmointi suoritettiin Microsoftin Visual Studio 2010 Ultimate -sovelluskehitysympäristössä.

Ohjelmoinnin aluksi tuli kartoittaa lopputuloksena syntyvän sovelluksen toimintaperiaatetta kattavasti ja suunnitella yksittäisten eri toimintojen sisältö, toteutustapa ja järjestys, jossa toiminnot tulisi suorittaa. Tietyn asteisen perusidean hahmottaminen ennen varsinaisen koodin kirjoittamista oli oleellista, jotta olisi olemassa työn kannalta suunnitelma ja suunta, joita seurata koko ohjelmointityön ajan. Lisäksi tuli pohjustaa, missä järjestyksessä sovelluksen eri toimintoja oli järkevää lähteä toteuttamaan ottaen huomioon kokonaisprojektiin sisältyvän integraation automaation kanssa.

Työn aluksi täytyi myös opetella valitun integrointitavan käyttämien tiedostojen XML-tiedostomuotoistandardia sekä CSV-tiedostomuotoisen datan periaatteita. Tämä oli oleellista sillä tiedostojen käyttötavat ja idea täytyi tiedonvälitysovellusta kehittäessä ymmärtää.

### 7.3.2 Ohjelmoinnin toteutus

Ohjelmointityö itsessään alkoi Rob-Ex-tuotannosuunnittelujärjestelmän käyttämien XML-muotoisten tiedostojen lukemisen ja päivittämisen luomisella, jonka jälkeen siirryttiin toteuttamaan valvomon ja sovelluksen välistä kommunikointia. Tämä siksi, että toisessa samanaikaisessa opinnäyteyössä kehitettävän valvomosovelluksen luominen alkoi samaan aikaan, joten tiedonvälitystä automaation kanssa ei olisi vielä pystytty käytännössä kokeilemaan. Sen sijaan Rob-Ex-ohjelmisto oli käyttövalmiina ja integraation testaus tuotannosuunnittelujärjestelmän kanssa oli mahdollista. Integrointi automaatioon sijoittui siis joka tapauksessa myöhempään ajankohtaan. Ohjelmointi toteutettiin rakentamalla kokonaisuovellusta osio kerrallaan aliohjelmina ja muokkaamalla eri toimintoja sitä mukaa, kuin lisää osia sovellukseen liitettiin.

Kaikkien osioiden tuli myös sisältää erilaisia lisäominaisuuksia, jotka luotiin osittain samanaikaisesti kutakin erillistä osaa ohjelmoitaessa ja osittain silloin, kun toimintoja lisättiin sovellukseen ja sovellusta laajennettiin. Lisäominaisuuksia vaadittiin sovellukseen kaikkien osioiden yhteensopivuuden mahdollistamiseksi.

Ohjelmakoodiin lisättiin kommentteja säännöllisesti työn aikana, jolloin koodin lukeminen ja kehittäminen helpottuivat huomattavasti. Kommentoitavia asioita olivat esimer-

kiksi eri aliohjelmien toiminnot eri vaiheissa ohjelmakoodia sekä erilaiset havaitut viat tai toimintansa kannalta keskeneräiset koodirivit. Kommenttien avulla oli helpompaa pysyä kartalla siitä, missä kohtaa ohjelmakoodia kehitettävää tai korjattavaa vielä oli.

Apua ohjelmointityöhön löytyi suurelta osin Internetistä valmiiden ohjelmakoodinäytteiden ja erilaisia toimintoja käsittelevien tutorial-artikkelien muodossa. Erityisesti Microsoftin Microsoft Developer Network -sivustosta oli huomattava apu ohjelmointikielen sisältämien valmiiden toimintojen toimintaperiaatteiden selvittämisessä.

### 7.3.3 Käytetyt menetelmät

Tietojen käsittelyä varten täytyi työkaluun ohjelmoida taulukkojärjestelmiä sovelluksen sisäisen tietojen tallentamisen ja muokkaamisen mahdollistamiseksi. Taulukkojärjestelmät koostuivat muuttujiksi määritellyistä erillisistä yksiulotteisista ja dynaamisista eli tarpeen mukaan kokoaan muuttavista taulukoista, joihin luettiin käytettävien resurssien, tilausten ja operaatioiden tiedot. Taulukot jaettiin ryhmiin siten, että luettuja resurssien, tilauksien sekä operaatioiden tietoja sisältäneet taulukot muodostuivat samankokoisiksi myöhempää tietojen käsittelyä suoraviivaistaen ja sitä kautta helpottaen. Yksiulotteisiin taulukoihin päädyttiin siitä syystä, että taulukoita on helpompaa käsitellä niiden kokoa muutettaessa, koska moniulotteisissa taulukoissa vain viimeisen sarakkeen kokoa voidaan muuttaa, mikäli aiempia taulukon sisältämiä tietoja halutaan säilyttää. Sovellukseen määritettiin lisäksi joukko julkisia sekä aliohjelmakohtaisia kokonaisluku-, merkkijono- sekä bittimuuttujia kuvan 13 mukaisesti, joita hyödynnettiin eri kohdissa sovellusta silmukkarakenteissa ja muissa ohjelmarakenteissa. [14]

```

'Importit:

Imports System.Xml
Imports System.IO
Imports System.Text

Public Class Form1

    'Yleiset muuttujat ja taulukot:

    Dim sTiedostopolkuReadRobex As String = ""
    Dim sTiedostopolkuWriteRobex As String = ""
    Dim TiedostopolutReadValvomo(0) As String 'Taulukoitu koska pitää pystyä lukemaan useita tiedostoja
    Dim sTiedostopolkuWriteValvomo As String = ""

    Dim iAikavali As Integer = 0
    Dim i As Integer = 0
    Dim n As Integer = 0
    Dim k As Integer = 0
    Dim p As Integer = 0
    Dim a As Integer = 0
    Dim q As Integer = 0
    Dim t As Integer = 0

    Dim TilausID() As String
    Dim Aloitus aika() As String
    Dim Aloitus aikaoperaatiot() As String
    Dim Asiakas() As String
    Dim Kpl() As String
    Dim Tila() As String
    Dim Tuote() As String
    Dim OperaatioID() As String
    Dim Operaatiot() As String
    Dim Operaatiotila() As String
    Dim Operaatioresurssinimi() As String
    Dim Operaatioresurssiryhma() As String
    Dim Resurssinimi() As String
    Dim Resurssityyppi() As String
    Dim Resurssiryhma() As String
    Dim Valmistuneet() As String
    Dim Hylatyt() As String

    Dim bOnOff As Boolean = False
    Dim bResurssitluettu As Boolean = False
    Dim bResurssiryhmat As Boolean = False
    Dim bKirjoitettavat_tilaukset As Boolean = False
    Dim bTilauksetluettu As Boolean = False
    Dim bOperaatioluettu As Boolean = False
    Dim bTiedostokirjoitettu As Boolean = False
    Dim bTiedostopolutOK As Boolean = False
    Dim bxmlMuuttunut As Boolean = False

    Dim bSetUpaikaloityty As Boolean = False
    Dim bcsvOperaatiotPilkku As Boolean = False

    Dim xmlTarkkailija As New FileSystemWatcher 'Määritellään luettavaa .xml -tiedostoa tarkkaileva toiminto.
    'Määritely tänne koska pitää päästä vaikuttamaan toimintaan useammasta aliohjelmasta käsin.

    'Määritellään koodausmuodot joita käytetään valvomolle kirjoitettavissa ja valvomolta luettavissa tiedostoissa.

    Dim Koodausmuoto As Encoding
    Dim KoodausmuotoUTF8 As New System.Text.UTF8Encoding(False) 'UTF-8 ilman BOM:ia
    Dim KoodausmuotoUTF8BOM As New System.Text.UTF8Encoding(True) 'UTF-8 ja BOM
    Dim koodausmuotoUTF32 As New System.Text.UTF32Encoding 'UTF-32
    Dim KoodausmuotoANSI As New System.Text.ASCIIEncoding 'ANSI eli ASCII
    Dim KoodausmuotoUNICODE As New System.Text.UnicodeEncoding 'UNICODE

    'Määritely tänne koska pitää päästä vaikuttamaan toimintaan useammasta aliohjelmasta käsin.

```

Kuva 13. Tiedonsiirtotyökalun yleiset muuttujat ja taulukot.

Käytettävien tiedostojen määrittämiseksi sovellukselle hyödynnettiin Visual Studio -sovelluskehitysympäristön sisältämiä valmiita OpenFileDialog- ja SaveFileDialog-toimintoja. Näiden avulla voidaan määrittää kuvan 14 mukaisesti avattavien tai tallennettävien tiedostojen tiedostopolut. Tiedostopolut tallennettiin muuttujiin ja niitä käytettiin sovelluksen eri osioissa tiedostojen lukemista ja kirjoittamista varten. [14]

```

'Rob-Ex:Ita luettavan .xml-tiedoston tiedostopolku:
OpenFD.InitialDirectory = "C:\\"
OpenFD.Title = "Valitse luettava xml -tiedosto:"
OpenFD.Filter = ".xml -tiedosto|*.xml"
OpenFD.FileName = ""
OpenFD.Multiselect = True
OpenFD.ShowDialog()

sTiedostopolkuReadRobex = OpenFD.FileName

```

Kuva 14. OpenFileDialog-toiminnon käyttö luettavan XML-tiedoston määrittämisessä.

XML-tiedostojen lukemisessa hyödynnettiin Microsoftin Visual Basic -ohjelmointikielen sisältämän System.xml-toimintokirjaston XmlReader-toimintoa. Tämän avulla voitiin lukea XML-muotoisesta tiedostosta aina halutut parametrit kustakin tiedoston sisältämästä elementistä tiedonvälitystä varten kuvan 15 mukaisesti. Taulukoihin lukeminen tapahtui XmlReader-toiminnon sisäistä while-silmukkarakennetta hyväksikäyttäen, ts. niin kauan kuin toiminto löysi luettavasta tiedostosta tietoja, suoritettiin komennot, jotka puolestaan poimivat halutut parametrit kullekin osoitettuihin taulukoihin, samalla taulukoiden kokoa kasvattaen.

```

Using xmltilauslukija As XmlReader = XmlReader.Create(sTiedostopolkuReadRobex)
While xmltilauslukija.Read()

    If xmltilauslukija.IsStartElement() Then

        If xmltilauslukija.Name = "ProdOrder" Then

            Dim sTilausnro As String = xmltilauslukija("name")
            ReDim Preserve TilausID(i)

            Dim sAloituspvm As String = xmltilauslukija("startCal")
            ReDim Preserve AloitusAika(i)

```

Kuva 15. XmlReader -toiminnon käyttö ja arvojen lukeminen muuttujiin.

Tietojen kirjoittaminen valvomolle tapahtui poimimalla tarvittavat tiedot taulukoista yksitellen ja kirjoittamalla ne sitten sovellukselle määritettyyn, valvomolle suunnattuun CSV-tiedostoon. Tietojen kirjoittamiseen hyödynnettiin Microsoftin System.IO-toimintokirjaston sisältämää StreamWriter-toimintoa, jonka avulla pystyttiin sovellukselle määritettyyn tiedostoon kirjoittamaan pilkuilla erotetut arvot kustakin taulukosta for-silmukkarakennetta hyödyntäen kuvan 16 mukaisesti. Tällöin kaikki tilaukset ja muut

tiedot oli mahdollista kirjoittaa tiedostoon riveittäin allekkain ja selkeässä järjestyksessä.

```
Dim ValvomoWrite As IO.StreamWriter = My.Computer.FileSystem.OpenTextFileWriter(sTiedostopolkuWriteValvomo, False)

For g = 0 To i - 1

    'Kirjoitetaan valvomolle vain 'Start Possible', 'Started' tai 'Paused' -tiloissa olevat tilaukset

    If Tila(g) = 40 Or Tila(g) = 50 Or Tila(g) = 60 Then

        'Operaatioiden kirjoitus tilausten alle

        If CheckBox2.Checked = True Then
            ValvomoWrite.WriteLine(TilausID(g) + "," + Aloitus aika(g) + "," + Asiakas(g) + "," + Tuote(g) + "," + Kpl(g))
        End If
    End If
End For
```

Kuva 16. StreamWriter-toiminnon käyttö tilausten kirjoittamisessa valvomolle

Tietojen lukeminen valvomolta käsin tapahtui käyttämällä Microsoftin TextFieldParser -nimistä toimintoa, joka mahdollisti tietojen lukemisen tekstitiedostosta toiminnon sisäiseksi taulukoiduksi dataksi. Valvomon kirjoittaman CSV-muotoisen tiedoston lukemiseksi täytyi toiminnolle määrittää tiedot erottelevaksi merkiksi pilkku kuvan 17 mukaisesti, jonka jälkeen toiminto osasi erotella tiedot pilkkujen väleistä omiin sarakkeisiinsa. Näistä toiminnon sisäisistä taulukon sarakkeista päivitettiin arvot sovelluksen omiin taulukoihin tiettyjen ehtojen puitteissa, jotta ne päätyisivät aina oikeisiin paikkoihin. Tiedot luettiin valvomon kirjoittamasta CSV-tiedostosta kaikilta mahdollisilta riveiltä while-silmukkarakennetta hyväksikäyttäen. Niin kauan kuin dataa tiedostosta löytyi, suoritettiin tietojen päivitystä sovelluksen sisäisiin taulukoihin määritettyjen ehtojen mukaisesti.

```
Using ValvomoRead As New Microsoft.VisualBasic.FileIO.
TextFieldParser(TiedostopoluReadValvomo(b))

ValvomoRead.TextFieldType =
    Microsoft.VisualBasic.FileIO.FieldType.Delimited
ValvomoRead.SetDelimiters(",")
```

Kuva 17. TextFieldParser-toiminnon ja välimerkkien määrittäminen

Tietojen kirjoitus XML-tiedostoon Rob-Ex-tuotannosuunnittelujärjestelmälle luettavaksi toteutettiin System.xml-toimintokirjaston sisältämien XML-tiedostojen käsittelyä koskevien komentojen avulla. Tietojen päivitys toimi siten, että tuotannosuunnittelujärjestelmältä luettavaksi saatu tiedosto avattiin ja sinne kirjoitettiin for-silmukkarakenteen avulla ja ohjelmalle määriteltyjen ehtolauseiden perusteella oikeisiin paikkoihin kunkin elementin attribuuttien arvoiksi sovellukseen taulukoidut päivitettyt tilaustiedot kuvan 18

mukaisesti. Tietojen päivittämisen lisäksi täytyi tiedostoon kirjoittaa uusi elementti ilmaisemaan tilauksen valmistuneiden ja hylättyjen kappaleiden lukumäärää. Sovelluksen suoritettua tietojen päivittämisen ja lisäämisen XML-tiedostoon tallennettiin tiedosto lopuksi Rob-Ex:lle luettavaksi tarkoitetun ja sovellukselle määritetyn tiedoston tiedostopolun mukaiseen kohteeseen. Tällöin tiedot päivittyivät alkuperäisen Rob-Ex:n kirjoittaman XML-tiedoston pohjalta luotuun kopioon, jonka Rob-Ex pystyi varmuudella lukemaan oikein.

```
Dim OmaXmlWRoot As XmlNode

OmaXmlWRoot = OmaXmlW.DocumentElement

'Käydään xml -tiedosto läpi ja päivitetään tiedot sekä lisätään <Progress> -tagit ja päivitetään nekin
p = 0
q = 0

For Each OmaXmlWElement In OmaXmlWRoot.ChildNodes

    For Each OmaXmlWChild In OmaXmlWElement.Childnodes

        If OmaXmlWChild.Name = "ProdOrder" Then

            OmaXmlWChild.Attributes("state").Value = Tila(p)
```

Kuva 18. XML-tiedostoon kirjoittaminen.

Koko päivityssyklin ajamiseksi eli aliohjelmien kutsumiseksi järjestyksessä, ohjelmoitiin sovellukseen kaksi tapaa. Ohjelmat ajetaan läpi joko sovellusikkunaan määritettyä nappulaa painamalla tai asetetaan ajettavaksi automaattisesti hyödyntäen sovelluskehitysympäristöstä löytyvää valmista Timer-toimintoa. Vaihtoehtoina on ajaa päivityssykli ajastimeen määritetyllä tasaisella aikavälillä tai tarkistaa ajastimen aikaväleillä, onko luettava XML-tiedosto muuttunut, ja ajaa päivityssykli läpi, mikäli näin on tapahtunut. Ajastimen aikavälin määrittämiseksi tehtiin sovellusikkunan "Asetukset"-välilehdelle kontrollit ja asetustäppä sekä niihin liittyvää ohjelmakoodia, joiden avulla haluttu toimintatapa voidaan määrittää. Tiedoston muuttumisen havaitsemiseksi käytettiin System.IO-toimintokirjaston sisältämää FileSystemWatcher-toimintoa kuvan 19 mukaisesti, jonka avulla kohdekansiossa sijaitsevaan tiedostoon tehdyt muutokset voitiin havaita ja aliohjelmasykli liipaista bittimuuttujan avulla.

```
'Tarkistaa onko luettava .xml -tiedosto päivittynyt entisestä kohdekansiossa.
```

Try

```
xmlTarkkailija.Path = Path.GetDirectoryName(sTiedostopolkuReadRobex)
xmlTarkkailija.NotifyFilter = (NotifyFilters.LastWrite)

AddHandler xmlTarkkailija.Changed, AddressOf Muuttunutxml
xmlTarkkailija.EnableRaisingEvents = True
```

Kuva 19. FileSystemWatcher -toiminnon käyttö

Muita lisäasetuksia, joita sovellusikkunaan luotiin, oli esimerkiksi mahdollisuus lukea ja kirjoittaa tietoja useamman eri valvomon kanssa, jota käytettäessä sovellus luo eri valvomoille suunnatut tiedostot tuotannosuunnittelujärjestelmään määritettyjen resurssiryhmän tietojen perusteella. Lisäksi sovellus pystyy lukemaan tiedot useammasta kuin yhdestä valvomon kirjoittamasta CSV-tiedostosta. Lisäasetuksiin kuuluivat myös kunkin tuotteen valmistamiseen vaadittavien operaatioiden kirjoittaminen valvomolle suunnattuun CSV-tiedostoon tilaustietoja seuraavalle riville, sekä valvomon tietojen lukeminen ja päivittäminen Rob-Ex-tuotannosuunnittelujärjestelmään operaatiokohtaisesti. Sovellukseen tehtiin myös mahdollisuus valita kirjoitettavan CSV-tiedoston merkkien koodaus. Näiden lisäasetusten ohjelmointi toteutettiin laajentamalla alkuperäisiä aliohjelmiä ja lisäämällä niihin tarvittavia ehtoja oikean toiminnan takaamiseksi. Lisäksi täytyi määrittää laajennetut tiedostomuodot sovelluksen ja valvomon välistä tiedonvälitystä varten.

#### 7.4 Ohjelmarakenne

Sovelluksen ohjelmakoodi rakentui aliohjelmista, jotka käyttivät yleisiä sekä paikallisia muuttujia tarpeen mukaan. Koko ohjelmakoodi kirjoitettiin yhteen moduuliin, koska ohjelmien välillä tuli toimintoja referoida muuttujiin, joita muut aliohjelmat kokonaisuudessa käyttivät. Koko ohjelmakoodin ollessa samalla ruudulla helpottui aliohjelmien selaaminen ja keskinäisten riippuvuuksien rakentaminen ilman ristiriitoja koodissa. Aliohjelmia oli yhteensä 9 kappaletta, ja ne ajettiin kutsumalla jokaista kerran tai vaihtoehtoisesti asetettiin ajettavaksi automaattisesti kuvan 20 mukaisesti. Aliohjelmat ajettiin aina samassa määritellyssä järjestyksessä seuraavasti:



- tiedostopolkujen määrittäminen sovellukselle
- resurssien lukeminen Rob-Ex:n kirjoittamasta XML-tiedostosta
- tilausten lukeminen Rob-Ex:n kirjoittamasta XML-tiedostosta
- operaatioiden lukeminen Rob-Ex:n kirjoittamasta XML-tiedostosta
- valvomolle CSV-muotoisen tilaustiedoston kirjoittaminen
- valvomon CSV-muotoisen päivitettyjä tilaustietoja sisältävän tiedoston lukeminen
- sovellusikkunassa olevaan tekstikenttään tilaustietojen kirjoittaminen
- päivitetyn XML-tiedoston kirjoittaminen Rob-Ex tuotannosuunnittelujärjestelmälle.

```
If bOnOff = True Then

    Tekstin_koodaus()
    Tarkista_luettava_xml()

    Label10.Text = "Työskentelee"

    Luexmlresurssit()
    Luexmltilaukset()
    Luexmloperaatiot()
    Kirjoitavalvomolle()
    Luevalvomolta()
    Taytataruutu()
    Kirjoitaxml()

    bxmlMuuttunut = False
    xmlTarkkailija.EnableRaisingEvents = False

    Label10.Text = "Valmis"
End If
```

Kuva 20. Aliohjelmien kutsuminen järjestyksessä.

Aliohjelmat ajettiin lävitse aina, mikäli niiden suorittamiseen määritetyt ehdot kutsumisen lisäksi täyttyivät. Kaikkien aliohjelmien ajamisessa oli yleisehtona, että sovellus oli määritelty "On"-tilaan. Tiedostopolkujen määrittämiseen tarkoitettu aliohjelma ajetaan läpi vain, jos jotakin tiedostopolkua ei ole määritelty. Tällöin tiedostopolkujen tulee olla aina määriteltyinä ennen muiden aliohjelmien ajamista. Eri tietojen lukemista koskevat aliohjelmat ajetaan lävitse vain, jos tiedostopolkujen määrittäminen oli onnistunut ja edellinen aliohjelma oli suoritettu onnistuneesti kuvan 21 mukaisesti.

```
Sub Luexmltilaukset()
    'Xml -tiedoston lukeminen ja tilaustietojen taulukointi

    If bOnOff = True And bResurssitluettu = True Then
```

Kuva 21. Aliohjelman ajamiseksi vaadittu ehtolause.

Valvomolle CSV-tiedoston kirjoittava aliohjelma suoritetaan, jos tiedot on luettu Rob-Ex:n kirjoittamasta XML-tiedostosta onnistuneesti. Tämän jälkeen valvomon kirjoittaman päivitettyt tilaustiedot sisältävän CSV-tiedoston lukeva aliohjelma ajetaan, mikäli tiedostot oli kirjoitettu valvomolle onnistuneesti. Taulukkoon tilaustiedot lisäävällä aliohjelmalla ei ole lisäehtoja kaikille yhteisen ehdon lisäksi. Lopuksi päivitettyt tiedot Rob-Ex-tuotannonsuunnittelujärjestelmälle XML-tiedostoon kirjoittava aliohjelma ajetaan läpi, mikäli tiedot valvomolta luettiin onnistuneesti.

Näiden aliohjelmien lisäksi sovelluksessa käytettiin myös muita pienempiä aliohjelmiä, joita kutsuttiin esimerkiksi tiedostopolkua tai ajastimen aikaa asetuksista vaihdettaessa. Näissäkin hyödynnettiin aliohjelmien kutsumismetodia, mutta erillisiä suorittamisehtoja ei määritelty.

## 7.5 Integrointi Rob-Ex:n ja WinCC:n välillä

Järjestelmän integrointi, eli tiedonsiirto, toteutettiin Rob-Ex:n ja WinCC:n välillä luomalla Rob-Ex-serverikoneelle jaettu kansio, jonka kautta valvomon tiedostoja pystyttiin kirjoittamaan ja lukemaan verkon ylitse. Kansioon päästiin käsiksi ottamalla yhteys Windowsin Käynnistä-valikon kautta valvomokoneelta serverikoneelle ja syöttämällä

serverikoneen käyttäjätunnus ja salasana valvomokoneen niitä kysyessä. Tämän jälkeen jaettu kansio määriteltiin valvomokoneelle käytön helpottamiseksi.

## 7.6 Testaus

Sovelluksen testausta suoritettiin koko ohjelmointiprosessin ajan tasaisin väliajoin. Koska integrointimenetelmäksi valittiin tiedostopohjainen tiedonsiirto, oli ohjelmaa mahdollista testata manipuloimalla tiedostoja ja ajamalla ohjelmaa läpi. Testausmenetelmiin kuuluivat lähinnä erilaiset kokeilut ja testiajot ohjelmakoodin parametreja sekä rakennetta muokkaamalla. Integrointia Rob-Ex-tuotannosuunnittelujärjestelmän kanssa testattiin automaatiolaboratoriossa ajamalla valmista kehitysversiota ohjelmasta projektikoneella ja tutkimalla, kuinka järjestelmät toimivat keskenään. Ongelmia havaittiin moneen otteeseen testien aikana, ja ne tarjosivat arvokasta tietoa seuraavien kehitysvaiheiden saralta. Testien perusteella siis selvisi, mitä tuli korjata ja mitä uutta kehittää järjestelmän toiminnan optimoimiseksi.

Integraatiota valvomon kanssa testattiin kokeilemalla tilaustietojen siirtämistä valvomolle ja lukemalla valvomon kirjoittamia tietoja takaisin Rob-Ex:ään. Testattaessa toimintaa ilmeni aluksi ongelmia valvomolle suunnattujen CSV-tiedostojen koodauksen kanssa. Valvomosovellus edellytti, että tiedot olisivat koodattuna ANSI-standardin mukaisesti ASCII-merkkeinä. Tiedonvälitystyökalu kirjoitti tiedot UTF-8 BOM-koodauksella, jolloin tiedostojen merkkijonojen eteen sijoittuva BOM (Byte Order Mark) sekoitti valvomosovelluksen tietojen lukemisen. Valvomosovellukseen lisättiin tämän seurauksena ominaisuus, jossa voitiin valita, mitä koodausta valvomolle kirjoitettavissa tiedostoissa käytettäisiin. Tällöin yhteensopivuutta erilaisten valvomosovellusten kanssa voitiin parantaa mahdollisen jatkokehityksen ja järjestelmän laajentamisen kannalta.

## 8 Yhteenveto

### 8.1 Aikaansaatu kokonaisuus

Opinnäytetyön tuloksena syntyi kokonaisuus, jonka avulla tilauksia ja tuotannon aika-  
taulutusta voidaan suunnitella Rob-Ex-tuotannosuunnittelujärjestelmässä ja tilaustie-  
toja välittää tiedonvälitystyökalun avulla tuotantolaitteistoa ohjaavan valvomosovelluk-  
sen ja Rob-Ex:n välillä. Käytössä oleva laboratorion automaatiolaitteisto mallinnettiin  
Rob-Ex-tuotannosuunnittelujärjestelmään ja sen käyttämistä käsittelevät harjoitustehtä-  
vät luotiin opetuskäyttöä varten. Tiedonsiirron mahdollistamiseksi tehtiin lisäksi tie-  
donvälitystyökalu käyttöohjeineen, jonka avulla Rob-Ex voidaan liittää automaatiota  
ohjaaviin valvomosovelluksiin tiedostopohjaista tiedonsiirtoa hyödyntäen.

### 8.2 Vertailu tavoitteisiin

Tämän opinnäytetyön tavoitteet saavutettiin kaiken kaikkiaan melko hyvin ja osapuil-  
leen aikataulussa. Tavoite selvittää erilaisten integrointimenetelmien käyttömahdollii-  
suudet ja toteutuksensa kannalta edullisimman menetelmän löytäminen täyttyivät. Li-  
säksi teoriapohjaisen tutkimuksen kannalta saatiin selvyttä siihen, miten erilaiset tie-  
donsiirtotekniikat toimivat ja miten standardi ISA-95 mallintaa tiedonsiirtoa tuotantoa  
harjoittavassa yrityksessä.

Käytännön tavoite saada aikaiseksi toimiva tiedonsiirto automaatiota ohjaavan valvo-  
mosovelluksen ja Rob-Ex-tuotannosuunnittelujärjestelmän välillä täyttyi ja Rob-Ex  
onnistuttiin mukauttamaan sopivaksi automaatiolaboratorion tarpeita varten. Lisäksi  
opetuskäyttöön tarkoitetut harjoitustehtävät saatiin luotua ja järjestelmän laajennus-  
mahdollisuudet ja menetelmät tämän opinnäytetyön näkökulmasta onnistuttiin selvittä-  
mään.

### 8.3 Kehitysmahdollisuudet tulevaisuudessa

Järjestelmän kehittäminen tulevaisuudessa on mahdollista laajentamalla valmistuk-  
senohjausjärjestelmää lisäämällä Rob-Ex-tuotannosuunnittelujärjestelmän yhteyteen  
historiatietoja keräävän tietokannan ja OEE-analysointijärjestelmän. Historiatietokanta

voidaan toteuttaa Historian-tietokannan avulla ja OEE-laskenta suoritetaan Proficy Plant Application -ohjelmistolla, johon on mahdollista konfiguroida vikapuut ja muut laskennalliset toiminnot. Näistä tietoja voidaan siirtää myös Rob-Ex:ään esimerkiksi SQL- tai XML-rajapintojen kautta.

Muita kehitysmahdollisuuksia tarjoaa Rob-Ex-tuotannosuunnittelujärjestelmän integrointi useampien laboratorion automaatiolaitteistojen kanssa. Tämä onnistuu hyödyntämällä tiedonvälitysohjelmiston lisäasetuksia, jolloin tilaustietoja voidaan jakaa useammille automaatiolaitteille ohjaaville valvomosovelluksille. Laajennuksen toteuttamiseksi laitteistoille tulisi kuitenkin luoda asianmukaiset ja tiedonvälitysohjelmiston kanssa yhteensopivat valvomosovellukset.

## Lähteet

- 1 Ahlroth, Harri. 2013. Asiantuntija, Novotek Oy, Vantaa. Palaveri 26.3.2013.
- 2 Gifford, Charlie. 2007. The Hitchhikers Guide to Manufacturing Operations Management ISA 95 Best Practices Book 1.0. ISA (Instrumentation Systems and Automation Society).
- 3 Heikniemi, Jouni. Mikä on XML? <http://www.heikniemi.fi/kirj/moxml.html>. Luettu 7.5.2013.
- 4 Hovi, Ari. 2004. SQL Opas. Saarijärven Offset Oy.
- 5 IANA. Internet Assigned Numbers Authority. <http://www.iana.org/>. Luettu 10.5.2013.
- 6 IETF. Common Format and MIME Type for Comma-Separated Values (CSV) Files. <http://tools.ietf.org/html/rfc4180>. Luettu 10.5.2013.
- 7 ISA. About ISA. [http://www.isa.org/Content/NavigationMenu/General\\_Information/About\\_ISA1/About\\_ISA.htm](http://www.isa.org/Content/NavigationMenu/General_Information/About_ISA1/About_ISA.htm). Luettu 9.5.2013.
- 8 ISA. ISA95 -standardin tuotekuvaus. <http://www.isa.org/Template.cfm?Section=Standards8&template=/Ecommerce/ProductDisplay.cfm&ProductID=11366>. Luettu 9.5.2013.
- 9 Iwanitz, Frank & Lange, Jürgen. 2006. OPC Fundamentals Implementation & Application. Heidelberg & Hüthig, cop.
- 10 Kippo, Asko K & Tikka, Aimo. 2008. Automaatiotekniikan perusteet. Edita.
- 11 Kletti, Jürgen. 2007. Manufacturing Execution Systems - MES. Springer-Verlag Berlin Heidelberg.
- 12 Kuikka, Seppo & Salmenperä, Mikko. 2004. 7601050 Tuotannon informaatiojärjestelmien integrointi -kurssin luentomateriaali. Tampereen teknillinen yliopisto. <http://www.ac.tut.fi/aci/courses/7601050/TIJKalvot2S2004.pdf>. Luettu 9.5.2013.
- 13 Liljaniemi, Antti. 2012. Tuotantoautomaatio-opintojakson luentomateriaali. Metropolia Ammattikorkeakoulu.

- 14 Microsoft Developer Network. Erilaiset ohjelmointiesimerkit ja toimintokuvaukset. <http://msdn.microsoft.com/en-US/>. Luettu 2.5.2013.
- 15 Novotek Planning Systems. 2009. Rob-Ex Gantt Manual.
- 16 OPC DataHub. What is OPC?.  
<http://www.opcdatahub.com/WhatIsOPC.html#note3>. Luettu 7.5.2013.
- 17 OPC Foundation. OPC Unified Architecture.  
[https://www.opcfoundation.org/Default.aspx/01\\_about/UA.asp?MID=AboutOPC#Specifications](https://www.opcfoundation.org/Default.aspx/01_about/UA.asp?MID=AboutOPC#Specifications). Luettu 7.5.2013.
- 18 OPC Foundation. What is OPC?  
[http://www.opcfoundation.org/Default.aspx/01\\_about/01\\_whatIs.asp?MID=AboutOPC](http://www.opcfoundation.org/Default.aspx/01_about/01_whatIs.asp?MID=AboutOPC). Luettu 7.5.2013.
- 19 Slack, Nigel & Chambers, Stuart & Johnston, Robert. 2007. Operations Management - fifth edition. Pearson Education.
- 20 Sustainability of Digital Formats Planning for Library of Congress Collections. CSV, Comma Separated Values.  
<http://www.digitalpreservation.gov/formats/fdd/fdd000323.shtml>. Luettu 10.5.2013.
- 21 W3C. Extensible Markup Language (XML) 1.0 Fifth Edition.  
<http://www.w3.org/TR/REC-xml/#dt-markup>. Luettu 7.5.2013.
- 22 W3C -organisaation kotisivut. <http://www.w3.org/>. Luettu 7.5.2013.
- 23 W3Schools. Webservices. <http://www.w3schools.com/webservices/>. Luettu 9.5.2013.

# **ReadWriter for Rob-Ex v1.0**

Tiedonvälitystyökalu

Käyttöohje



## Sisältö

1. Yleistä	3
2. Asennus	3
3. Käyttöliittymä	5
3.1 Tilaukset-välilehti	5
3.2 Asetukset-välilehti	6
3.3 Ylävalikot	7
4. Toiminta	8
4.1 Toimintaperiaate	8
4.2 Erilaiset toimintamahdollisuudet	9
5. Ohjelman käyttö	11
5.1 Käyttösekvenssi	11
5.2 Tiedostopolkujen määrittäminen	11
5.3 Tietojen päivitys	12
6. Välitettävät tiedot	12
7. Valvomon tiedostomuodot	14
7.1 Työkalun kirjoittamat tiedostot	14
7.2 Työkalun lukemat tiedostot	15
7.3 Tiedostojen sijainti	18
8. Valvomon asetukset	18
8.1 Tietojen lukeminen	18
8.2 Tietojen kirjoitus	19
9. Rob-Ex-asetukset	19
9.1 Tietojen kirjoittaminen	19
9.2 Tietojen lukeminen	21
9.3 Resurssiryhmät	23

## 1. Yleistä

Ohjelma on luotu välittämään tietoa Rob-Ex -valmistuksensuunnittelujärjestelmän sekä automaatiota ohjaavan valvomosovelluksen välille. Tiedonvälitys tapahtuu työkalun avulla tiedostopohjaisesti Rob-Ex:n sekä valvomon lukemien ja kirjoittamien tiedostojen lukemisella ja niiden päivittämisellä.

Lisätietoa ohjelman toiminnasta sekä projektista, jonka tuloksena ohjelma syntyi, löytyy Metropolia Ammattikorkeakoululle tehdystä insinööriyöstä: Rob-Ex-tuotannonsuunnittelujärjestelmän integrointi - Antti Ruohio - 2013.

***Huom! Tämä ohjelma ei ole virallinen tuotos, joten kaikkeen se ei valitettavasti pysty.***

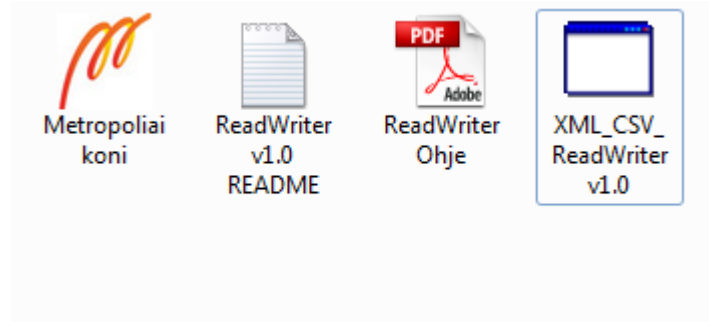
## 2. Asennus

Ohjelma voidaan asentaa tarpeeksi uudella Windows-käyttöjärjestelmällä varustettuun koneeseen. Asennus voidaan suorittaa asennustiedostosta, jos sellainen löytyy, tai vaihtoehtoisesti suoraan ohjelman kansion tietokoneelle kopioimalla.

Ohjelma ei sisällä erillisiä elementtejä, jotka olisivat toiminnan kannalta oleellisia, vaan se voidaan ajaa suoraan .exe-tiedostosta. Kansiota tulisi kuitenkin löytyä seuraavat:

- Pikakuvakkeen ikoni (.ico)
- ReadMe-tiedosto (.txt)
- Ohjelman ajotiedosto(.exe)
- Ohje(.pdf).

Ohjelman ajotiedostosta suositellaan luotavan pikakuvake, jolle on tunnusomaista kansiota löytyvä ikoni sekä sijainti työpöydällä.

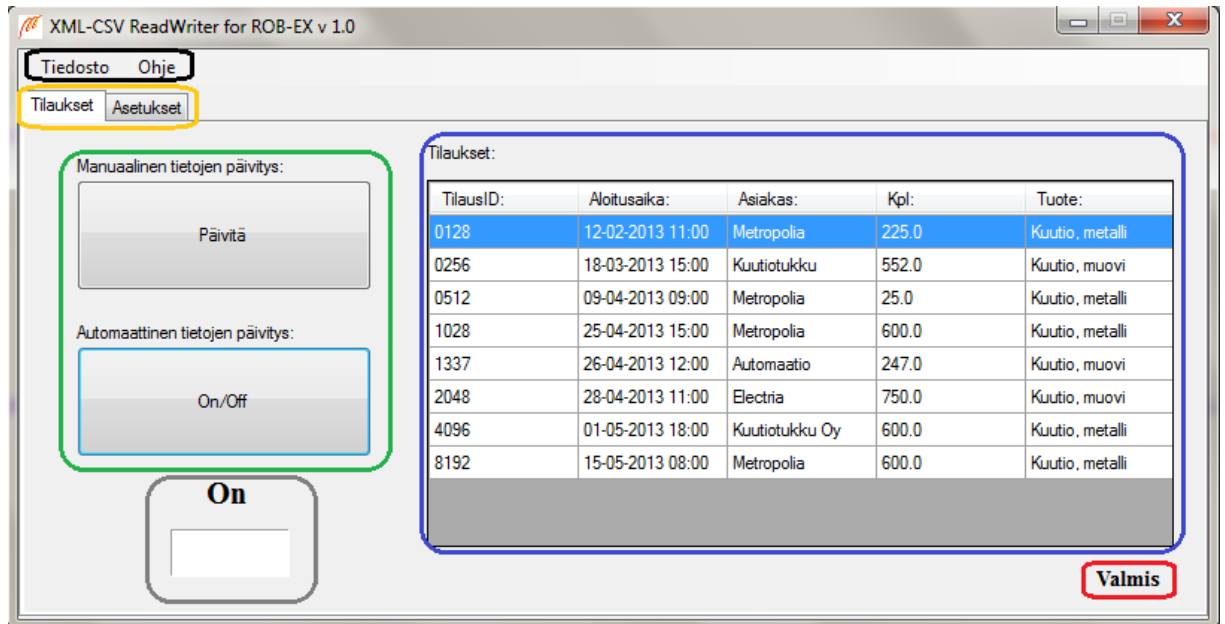


*Kuva 1: Ohjelman kansion sisältämät tiedostot*

Ohjelma suositellaan asennettavaksi samalle serverikoneelle, jolle Rob-Ex on asennettu. Tällöin työkalu voi toimia Rob-Ex:n kyljessä samanaikaisesti ja jakaa tietoja verkon yli tiedostoina oikeille kohteille ts. valvomoille luettaviksi. Työkalua voi käyttää myös muualta niin kauan, kuin tiedostoja voi lukea ja kirjoittaa verkon ylitse.

### 3. Käyttöliittymä

#### 3.1 Tilaukset-välilehti



Kuva 2: Tilaukset välilehti

- **Ylävalikot**

-Ylävalikot, joista löytyvät ohjelman toiminnot ohjelmaikkunaan sijoitettujen kontrollien lisäksi.

- **Välilehden valinta**

-Kontrollit välilehtien "Tilaukset" ja "Asetukset" välillä siirtymiseksi.

- **Tietojen päivittäminen manuaalisesti tai automaattisesti**

-Painonapit, joita painamalla voidaan tiedot päivittää Rob-Ex:n ja valvomon välillä kertapäivityksellä tai asettaa päivitettäväksi automaattisesti.

- **On/Off - valkoinen/musta**

-Ilmaisee, päivittääkö ohjelma tietoja Rob-Ex:n ja valvomon välillä.

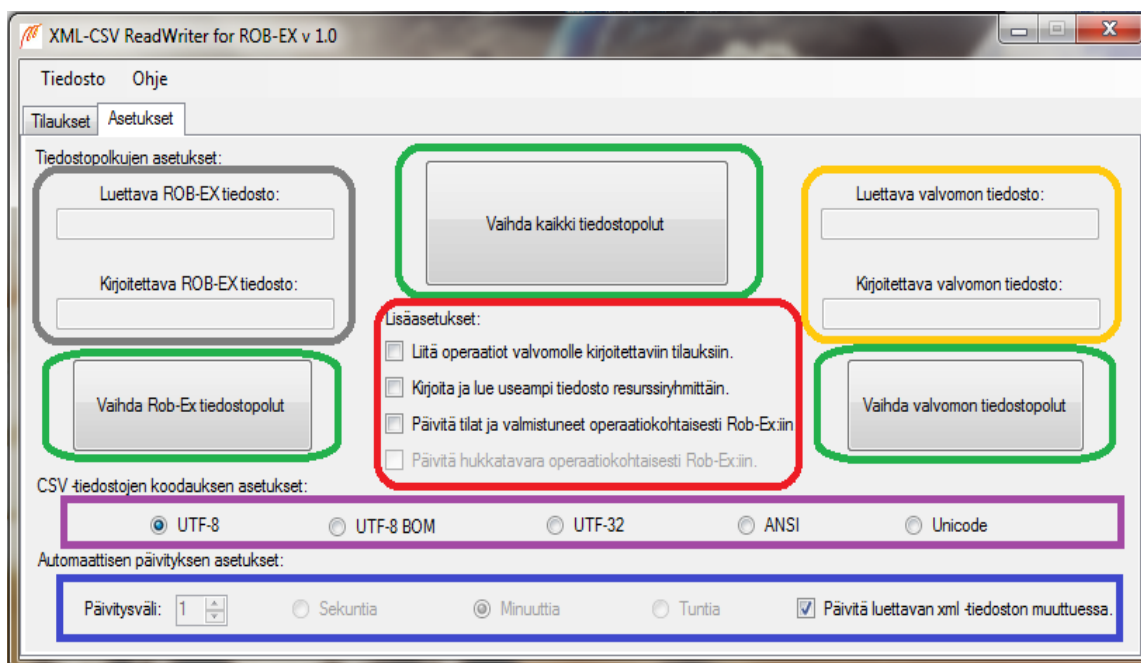
- **Tilausruudukko**

-Näyttää Rob-Ex:n kirjoittamasta XML -tiedostosta luetut tilaukset.

- **Ohjelman tila**

-Kertoo ohjelman hetkellisen tilan.

### 3.2 Asetukset-välilehti



Kuva 3: Asetukset välilehti

- **Tiedostopolut Rob-Ex**

-Näyttää määritetyt Rob-Ex:n tiedostojen lukemisessa ja kirjoittamisessa käytettävät tiedostopolut.

- **Tiedostopolut Valvomo**

-Näyttää määritetyt valvomon tiedostojen lukemisessa ja kirjoittamisessa käytettävät tiedostopolut.

- **Tiedostopolkujen vaihtaminen**

-Kontrollit kaikkien, Rob-Ex:n tai valvomon tiedostojen lukemisessa ja kirjoittamisessa käytettävien tiedostopolkujen vaihtamiseksi.

- **Automaattisen päivityksen asetukset**

-Kontrollit automaattisen päivityksen asettamiseksi toimimaan aikasyklillä tai luettavan XML-tiedoston päivittyessä kohdekansiossa.(Rob-Ex) Lisäksi kontrollit aikasyklin pituuden määrittämiseksi halutun pituiseksi.

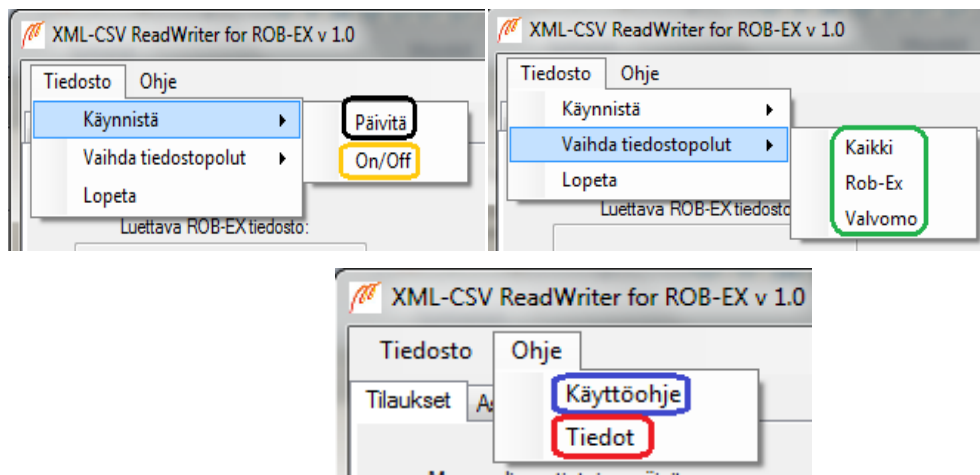
- **Lisäasetukset**

-Kontrollit lisäasetusten määrittämiseksi joiden avulla ohjelman toimintaa voidaan laajentaa perustoiminnoista. (kts. 4.2 Erilaiset toimintamahdollisuudet)

- **CSV -tiedostojen koodauksen asetukset**

-Kontrollit kirjoitettavien CSV-tiedostojen tekstin merkkien koodausmenetelmän valitsemiseksi.

### 3.3 Ylävalikot



Kuva 4: Ylävalikot

- **Tietojen päivittäminen manuaalisesti**

-Päivittää tiedot Rob-Ex:n ja valvomon välillä kerran

- **Automaattisen tietojen päivittämisen asettaminen**

-Asettaa automaattisen tietojen päivityksen päälle/pois

- **Tiedostopolkujen vaihtaminen**

-Vaihtaa kaikkien, Rob-Ex:n tai valvomon tiedostojen lukemisessa ja kirjoittamisessa käytettävät tiedostopolut.

- **Käyttöohjeen näyttäminen**

-Näyttää tämän käyttöohjeen.

- **Ohjelman tietojen näyttäminen**

-Näyttää ohjelman tiedot.

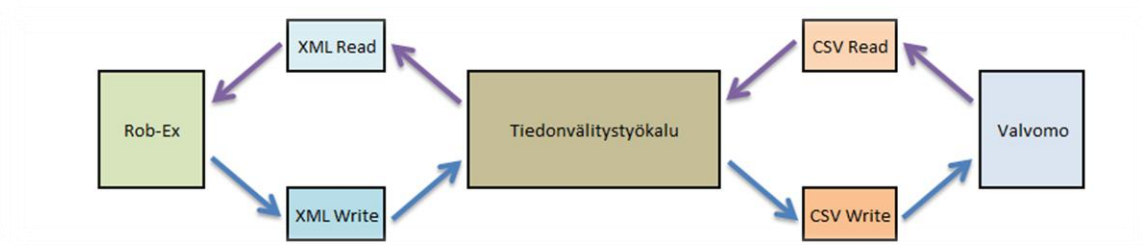
## 4. Toiminta

### 4.1 Toimintaperiaate

Ohjelma käyttää yhtä valvomoa kohden yhteensä neljää tiedostoa joita ovat:

- Rob-Ex:n tuottama tiedosto tilauksista (.xml)
- Rob-Ex:lle luettavaksi luotava tiedosto (.xml)
- Valvomolta luettava tiedosto (.csv)
- Valvomolle kirjoitettava tiedosto (.csv).

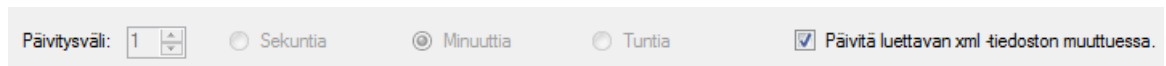
Mikäli tiedonsiirto määritetään toimivaksi useamman valvomon kanssa, on jokaista valvomoa kohti oltava oma luettava sekä kirjoitettava CSV -muotoinen tiedostonsa. Ohjelmalle tulee ensin määrittää kaikkien tiedostojen sijainti eli tiedostopolku, jonka jälkeen ohjelma päivittää vain olennaisia tilaus ja operaatietietoja valvomon ja Rob-Ex:n välillä. Ohjelma siis suodattaa ja välittää tietoja edestakaisin valvomon ja Rob-Ex:n välillä.



Kuva 5: Tiedonsiirron periaate

#### 4.2 Erilaiset toimintamahdollisuudet

Ohjelmaa voidaan käyttää joko manuaalisesti kertapäivityksellä tai se voidaan asettaa toimimaan automaattisesti aikasyklillä tai joko Rob-Ex:n kirjoittaman tiedoston päivityksessä. (Oletus)

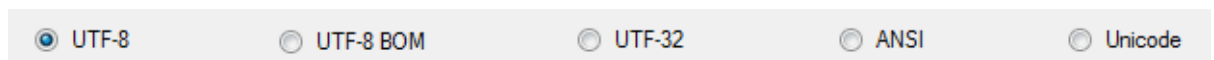


Kuva 6: Automaattisen päivityksen asetukset

**Huom! Aikasykli koostuu 5s -60s, 1min - 60min ja 1h -24h päivitysväleistä ja määritettävissä ovat vain kokonaiset aikayksiköt.**

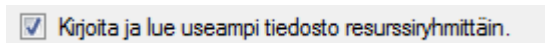
Valvomoille kirjoitettavien CSV-tiedostojen merkkijonojen koodaustapa voidaan valita UTF-8, UTF-8 BOM, UTF-32, ANSI ja Unicode-koodauksen välillä.

**Huom! Valinta pätee kaikkiin kirjoitettaviin CSV-tiedostoihin.**



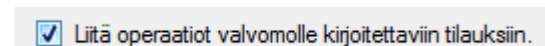
Kuva 7: CSV-tiedostojen koodauksen valinta.

Tiedonvälitys voidaan asettaa toimimaan yhden tai useamman valvomon ja sitä kautta automaattilaitteiston kanssa. Kirjoitettaessa tietoja useammalle valvomolle täytyy Rob-Ex:ään olla määriteltyinä jokaista laitteistoa kuvaavat resurssiryhmät.



Kuva 8: Useamman tiedoston kirjoitus ja lukeminen.

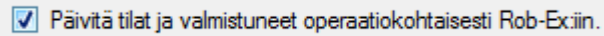
Tietojen päivitys valvomolle voidaan suorittaa joko pelkkien tilaustietojen tai tarvittaessa tilaustietojen sekä tilaukseen liittyvien operaatioiden kirjoittamisella valvomolle. Mikäli tilauksia on tarve jakaa useamman resurssiryhmän välillä tai tilausten reitti vaihtelee, on suositeltavaa ottaa käyttöön operaatioiden kirjoittaminen valvomolle tilausten yhteyteen.



Kuva 9: Operaatioiden kirjoittaminen valvomolle.



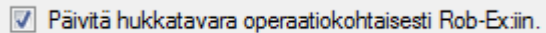
Tietojen päivittäminen Rob-Ex:ään voidaan tehdä tilaustasolla (Oletus) tai operaatiokohtaisesti tarpeen mukaan. Mikäli tilauksia on tarve jakaa useamman resurssiryhmän välillä, on otettava käyttöön tietojen päivitys Rob-Ex:ään operaatiokohtaisesti.



Päivitä tilat ja valmistuneet operaatiokohtaisesti Rob-Ex:iin.

*Kuva 10: Tietojen päivittäminen Rob-Ex:ään operaatiokohtaisesti.*

Lisäasetuksena operaatiokohtaisesti tietoja päivittäessä voidaan hukkatavaran määrä päivittää Rob-Ex:ään joko tilauskohtaisesti tai operaatiokohtaisesti.



Päivitä hukkatavara operaatiokohtaisesti Rob-Ex:iin.

*Kuva 11: Hukkatavaran päivitys Rob-Ex:ään operaatiokohtaisesti.*

**Huom! Tietojen päivitys operaatiokohtaisesti suoritettaessa täytyy valvomolta saada luettua operaatioista yksityiskohtaisempaa tietoa oikean toiminnan takaamiseksi. (kts. 7.2 Työkalun lukemat tiedostot)**

## 5. Ohjelman käyttö

### 5.1 Käyttösekvenssi

Ohjelman käyttö aloitetaan määrittämällä ensin ohjelmalle käytettävien tiedostojen sijainti eli tiedostopolut. Tämän jälkeen ohjelma voidaan asettaa joko automaattiajolle tai käyttää manuaalista kertapäivitystä. Tällöin ohjelma lukee ja kirjoittaa tietoja Rob-Ex:ltä valvomolle ja valvomolta Rob-Ex:lle.

Ohjelman voi käynnistää tupla klikkaamalla pikakuvaketta tai ohjelman ajotiedostoa. Sulkeminen tapahtuu ohjelman ikkunan oikeassa yläkulmassa olevasta ruksista tai valitsemalla ylävalikosta "Tiedosto→Lopeta".

***Huom! Ohjelma ei tallenna omia tietojaan kuten tiedostopolkuja ym. minnekään, joten ohjelmaa suljettaessa häviävät tiedot ja ne täytyy seuraavan kerran käynnistettäessä määrittää jälleen uudelleen.***

### 5.2 Tiedostopolkujen määrittäminen

Tiedostopolut voi määrittää Asetukset -välilehdellä painamalla "Vaihda Rob-Ex - tiedostopolut", "Vaihda valvomon tiedostopolut" tai "Vaihda kaikki tiedostopolut" - nappulaa tai valitsemalla ylävalikosta "Tiedosto→ Vaihda tiedostopolut→Kaikki/Rob-Ex/Valvomo". Tällöin ohjelma tyhjentää ko. tiedostopolut ja kysyy tilalle uusia.

Tiedostopolkujen määrittäminen tapahtuu myös, mikäli tietojen päivitystä yritetään käynnistää eikä kaikkia tiedostopolkuja ole määritetty. Tiedostopolkujen määrittämisen kesken jättäminen antaa virheilmoituksen. Määritetyt tiedostopolut näkyvät Asetukset - välilehdellä olevissa tekstikentissä.

Mikäli lisäasetus useiden tiedostojen lukemista ja kirjoittamista varten on valittuna, määritetään valvomolta luettavia tiedostoja useampi ja luodaan valvomolle kirjoitettavia tiedostoja useampi.

***Huom! Ohjelman toiminnan takaamiseksi tarvitaan vain luettavat tiedostot. Ohjelma luo Rob-Ex:lle ja valvomoille luettavaksi tarkoitetut tiedostot tiedostopolkuja määritettäessä.***

**Huom! Useita tiedostoja valvomolle kirjoitettaessa lisää ohjelma tiedostonimen perään kunkin resurssiryhmän nimen sekä tiedostopäätteen automaattisesti! Hyvä tallennuskunaan määritetty nimi tiedostolle olisi siis esim. "Valvomolle", josta muodostuu resurssiryhmän ollessa "Linjasto1" tiedosto nimeltä "ValvomolleLinjasto1.csv".**

### 5.3 Tietojen päivitys

Tietojen kertapäivityksen voi suorittaa painamalla "Päivitä" -nappulaa ohjelman Tilaukset -välilehdellä tai valitsemalla ylävalikosta "Tiedosto → Käynnistä → Päivitä". Tällöin ohjelma päivittää tiedot Rob-Ex:n ja valvomon välillä kerran.

Automaattiajolle ohjelman voi asettaa painamalla "On/Off" -nappulaa tai valitsemalla ylävalikosta "Tiedosto → Käynnistä → On/Off". Tällöin ohjelma päivittää tietoja Rob-Ex:n ja valvomon välillä automaattisesti sille määritettyjen asetusten mukaan. Automaattipäivityksen saa kytkettyä pois päältä valitsemalla "On/Off" uudelleen.

## 6. Välitettävät tiedot

Valvomolle välitettäviä tilaustietoja ovat:

- Tilauksen ID -numero
- Tilauksen tuottamisen aloitusajankohta
- Asiakkaan nimi
- Valmistettava kappalemäärä
- Tuotenimike

Valvomolle välitettäviä operaatiotietoja ovat:

- Operaation aloitusajankohta
- Operaation nimi
- Operaatiossa käytettävä resurssi (kone)

**Huom! Valvomolle välitetään vain tilaukset joiden tila on "Start Possible", "Started" tai "Paused". Lisäksi valvomolle välitetään operaatiotietoja vain jos lisäasetus operaatiotietojen liittämiseksi valvomolle on valittuna. Tällöin myös tiedostomuodot muuttuvat. (kts. 7.3 Valvomon tiedostomuodot)**

Valvomolta luettavia tilaustietoja ovat:

- Tilauksen ID -numero
- Tilauksen tilakoodi
- Hukkatavaran määrä (tilauskoht.)
- Tuotettujen lukumäärä (resurssikoht.)

Valvomolta luettavia operaatitietoja ovat:

- Operaation nimi
- Operaatiossa käytettävä resurssi (kone)
- Operaation tilakoodi
- Tuotettujen lukumäärä
- Hukkatavaran määrä

***Huom! Valvomolta luetaan operaatitietoja vain jos lisäasetus tietojen päivittämiseksi operaatiokohtaisesti Rob-Ex:lle on valittuna. Tällöin myös tiedostomuodot muuttuvat. (kts. 7.3 Valvomon tiedostomuodot)***

## 7. Valvomon tiedostomuodot

Seuraavassa määriteltynä minkälaisessa muodossa valvomolle kirjoitettavat ja valvomolta luettavat CSV-tiedostot tulee olla kirjoitettuina jotta niiden sisältämien tietojen lukeminen onnistuu oikein. Valvomon tulee siis osata lukea ja kirjoittaa tiedot oikeassa muodossa tarpeen mukaan.

### 7.1 Työkalun kirjoittamat tiedostot

- **Valvomolle, pelkät tilaukset (Oletus):**

0128,12-02-2013 11:00,Metropolia,Kuutio metalli,225.0
0256,18-03-2013 15:00,Kuutiotukku,Kuutio muovi,552.0
0512,09-04-2013 09:00,Metropolia,Kuutio metalli,25.0

*Tilaukset lueteltuina allekkain tiedostoon!*

**Joista riveillä järjestyksessä:**

*TilausID, Aloitusaika, Asiakas, Tuote, Valmistettava kappalemäärä (tilauskoht.)*

- **Valvomolle tilaukset ja operaatiot:**

0128,12-02-2013 11:00,Metropolia,Kuutio metalli,225.0
25-04-2013 15:00,Syöttö,Syöttöasema1,25-04-2013 15:05,Prässäys,Prässi,18-04-2013 08:13,Varastoon siirto,Varasto
0256,18-03-2013 15:00,Kuutiotukku,Kuutio muovi,552.0
25-04-2013 15:00,Syöttö,Syöttöasema2,25-04-2013 15:05,Prässäys,Prässi,18-04-2013 08:13,Varastoon siirto,Varasto

*Tilaukset ja seuraavalla rivillä tilausta koskevat operaatiot lueteltuina allekkain tiedostoon!  
Operaatiot lueteltu omalle rivilleen peräkkäin.*

**Joista ensimmäisellä rivillä tilaustiedot järjestyksessä:**

*TilausID, Aloitusaika, Asiakas, Tuote, Valmistettava kappalemäärä (tilauskoht.)*

**Toisella rivillä operaatioiden tiedot liittyen edellisen rivin tilaukseen:**

*Operaation aloitusaika, Operaation nimi, Resurssin nimi, Operaation aloitusaika, Operaation nimi, Resurssin nimi, Operaation aloitusaika, Operaation nimi, Resurssin nimi, jne...*

7.2 Työkalun lukemat tiedostot

● **Valvomolta, pelkät tilaukset (Oletus):**

0128,80,4.0,225.0,225.0,225.0
0256,50,2.0,321.0,320.0,319.0
0512,60,0.0,12.0,12.0,12.0

*Tilaukset lueteltuina allekkain tiedostoon!*

**Joista riveillä järjestyksessä:**

*TilausID, Tilauksen tila, Hukatavara(tilauskoht.), Tuotetut kone 1, Tuotetut kone 2, Tuotetut kone 3, jne..*

**Mahdolliset tilauksen tilakoodit:**

- 50=Started
- 60=Paused
- 80=Completed

**HUOM! Toimii usean valvomon kanssa vain jos tilaukset ovat kokonaan yhdelle valvomolle suunnattuja! ts. Tilauksia ei saa hajauttaa useammille laitteistoille/resurssiryhmille.**

● **Valvomolta, tilaukset ja operaatiot:**

0128,80,4.0 Syöttö,Syöttöasema1,70,225.0,Prässäys,Prässi,70,225.0,Varastoon siirto,Varasto,70,225.0 0256,50,2.0 Syöttö,Syöttöasema2,40,321.0,Prässäys,Prässi,40,320.0,Varastoon siirto,Varasto,40,319.0
--

*Tilaukset ja seuraavalla rivillä tilausta koskevat operaatiot lueteltuina allekkain tiedostoon!  
Operaatiot lueteltu omalle rivilleen peräkkäin.*

**Joista ensimmäisellä rivillä tilaustiedot järjestyksessä:**

*TilausID, Tilauksen tila, Hukkatavara(tilauskoht.)*

**Toisella rivillä operaatioiden tiedot liittyen edellisen rivin tilaukseen:**

*Operaation nimi, Resurssin nimi, Operaation tila, Tuotetut, Operaation nimi, Resurssin nimi,  
Operaation tila, Tuotetut, Operaation nimi, Resurssin nimi, Operaation tila, Tuotetut, jne..*

**Mahdolliset tilauksen tilakoodit:**

- 50=Started
- 60=Paused
- 80=Completed

**Mahdolliset operaatioiden tilakoodit:**

- 40=Started
- 50=Paused
- 70=Completed

**HUOM! Pitää käyttää silloin, kun tilauksia hajautetaan useammille valvomoi-  
le/resurssiryhmille!**

- **Valvomolta, tilaukset ja operaatiot sekä hukkatavara operaatiokoh-  
taisesti**

0128,80

Syöttö,Syöttöasema1,70,225.0,2.0,Prässäys,Prässi,70,225.0,2.0,Varastoon siirto,Varasto,70,225.0,0.0

0256,50

Syöttö,Syöttöasema2,40,321.0,0.0,Prässäys,Prässi,40,320.0,2.0,Varastoon siirto,Varasto,40,319.0,0.0

*Tilaukset ja seuraavalla rivillä tilausta koskevat operaatiot lueteltuina allekkain tiedostoon!  
Operaatiot lueteltu omalle rivilleen peräkkäin.*

**Joista ensimmäisellä rivillä tilaustiedot järjestyksessä:**

*TilausID, Tilauksen tila*

**Toisella rivillä operaatioiden tiedot liittyen edellisen rivin tilaukseen:**

*Operaation nimi, Resurssin nimi, Operaation tila, Tuotetut, Hukkatavara, Operaation nimi,  
Resurssin nimi, Operaation tila, Tuotetut, Hukkatavara, Operaation nimi, Resurssin nimi,  
Operaation tila, Tuotetut, Hukkatavara, jne..*

**Mahdolliset tilauksen tilakoodit:**

- 50=Started
- 60=Paused
- 80=Completed

**Mahdolliset operaatioiden tilakoodit:**

- 40=Started
- 50=Paused
- 70=Completed



### 7.3 Tiedostojen sijainti

Rob-Ex:n kirjoittama sekä Rob-Ex:lle luettavaksi tarkoitetut XML-tiedostot voidaan sijoittaa omaan kansioonsa tietokoneelle, jossa Rob-Ex Gantt multiuser -serveriä sekä tiedonvälitystyökalua ajetaan, tai muualle josta ne ovat luettavissa ja johon tietoa voidaan kirjoittaa. Kuitenkin tiedostot tulisi pitää omassa kansiossaan ilman muita tiedostoja.

Valvomoille tarkoitetut CSV-tiedostot tulisi sijoittaa esim. verkon yli jaettuun kansioon serverikoneelle tai muualle josta ne ovat luettavissa ja johon tietoa voidaan kirjoittaa. Luettavat ja kirjoitettavat tiedostot suositellaan sijoitettavaksi omiin kansioihinsa, mutta ne voidaan sijoittaa myös samaan kansioon. Tietoturvaa voidaan parantaa luomalla jaettuun kansioon säännöt käyttäjistä, jolloin ulkopuoliset eivät pääse tietoja manipuloimaan.

***Huom! Tiedostoja sisältäviin kansioihin tulee olla käyttöoikeudet!***

## 8. Valvomon asetukset

### 8.1 Tietojen lukeminen

Valvomon lukiessa tietoja tiedonvälitystyökalun kirjoittamasta .csv -tiedostosta, täytyy valvomolle olla määriteltynä oikeanlainen lukemissekvenssi. Tiedostossa näkyviä tietoja ovat valvomolle suunnatut tilaus ja/tai operaatiotiedot, joiden tilauksen tilaksi on määriteltty joko "Start Possible", "Started" tai "Paused". Esimerkiksi peruutetut tai valmiit tilaukset eivät siis valvomolle päin näy.

***Huom! Tiedostoja lukiessa on noudatettava oikeita tiedostomuotoja! (kts 7.1 Työkalun kirjoittamat tiedostot)***

***Huom! Tiedostojen koodauksen täytyy olla yhteensopiva valvomon ja tiedonvälitysohjelman kanssa.***

## 8.2 Tietojen kirjoitus

Valvomon tulisi kirjoittaa tilauksen tilatiedot työkalun luettavaksi tarkoitettuun tiedostoon vasta sitten, kun tilauksen tuottaminen on aloitettu. Valvomon tulee määrittää tilauksen tilakoodiksi 50 "Started" kun tilaus on aloitettu. Mikäli tilauksen tuottaminen keskeytyy, tulee valvomon määrittää tilauksen tilakoodiksi tiedostoon 60 "Paused". Mikäli tilaus tulee valmiiksi, tulee valvomon määrittää tilauksen tilakoodiksi 80 "Completed". Valvomon tulisi kuitata tilaus valmiiksi vasta kun tilauksessa määritetty kappalemäärä on tuotettu onnistuneesti.

Käytettäessä laajennettua toimintamoodia, jossa valvomolta luetaan tilaustietojen lisäksi operaatiokohtaiset tiedot, täytyy valvomon noudattaa samoja sääntöjä tilatietojen päivittämisessä kuin edellä paitsi että tilakoodit ovat 40 "Started", 50 "Paused" ja 70 "Completed".

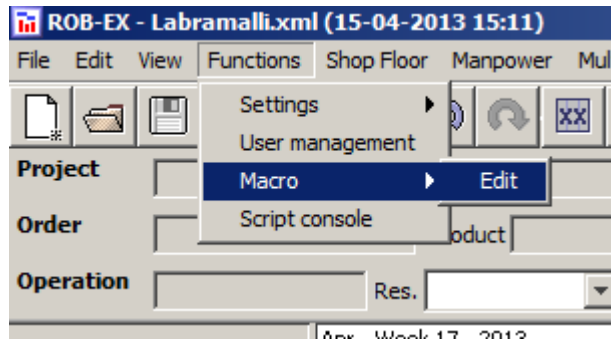
***Huom! Tiedostoja kirjoitettaessa on noudatettava oikeita tiedostomuotoja! (kts. 7.2 Työkalun lukemat tiedostot)***

***Huom! Tiedostojen koodauksen täytyy olla yhteensopiva valvomon ja tiedonvälitysohjelman kanssa.***

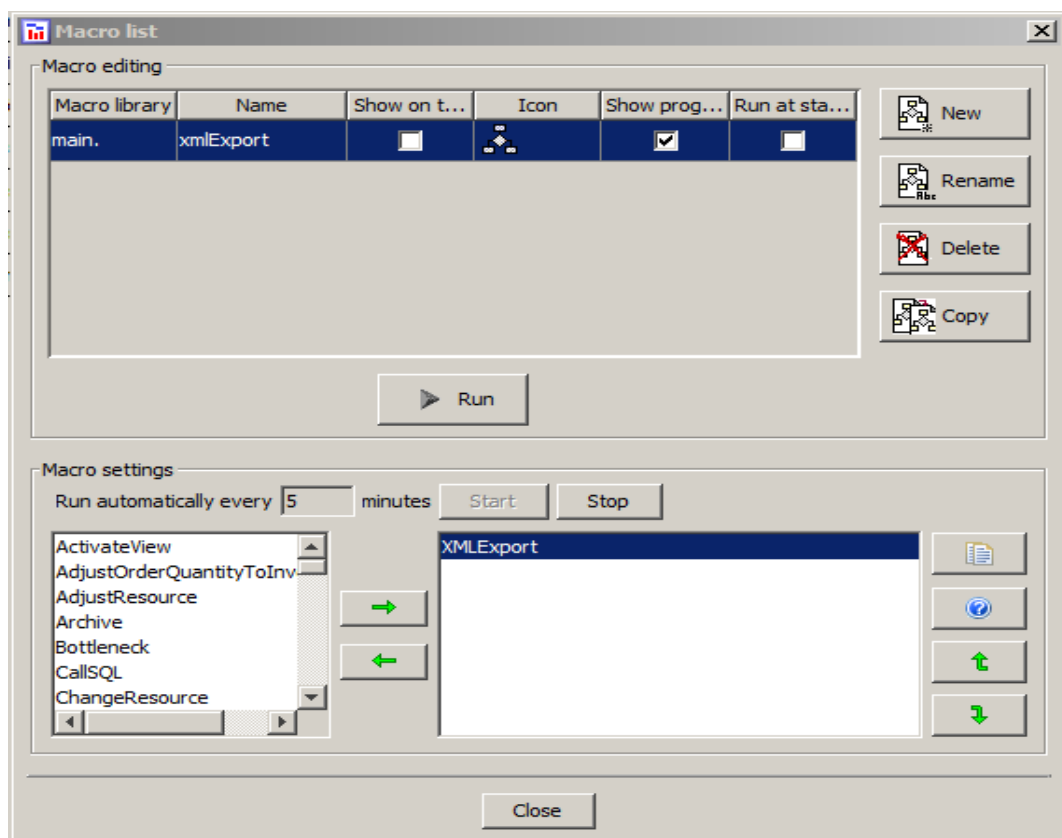
## 9. Rob-Ex-asetukset

### 9.1 Tietojen kirjoittaminen

Rob-Ex:n tilaustietojen kirjoittamisessa luettavaksi työkalulle hyödynnetään Rob-Ex:n valmiita makrotoimintoja, joiden avulla voidaan määrittää tilaustiedoista koostuva XML-tiedosto kirjoitettavaksi haluttuun kohdekansioon. Makrotoiminnot jäävät Rob-Ex:ään muistiin, joten ne ovat saatavilla heti seuraavallakin kerralla ohjelmaa käynnistettäessä. Makrotoimintona tietojen kirjoittamiseksi suositellaan käytettävän Rob-Ex:n valikosta valitsemalla "Functions→Macro→Edit " löytyvää makroeditoria, johon voidaan määrittää XML Export -niminen makrotoiminto.



Kuva 12: Rob-Ex:n makroeditorin valinta

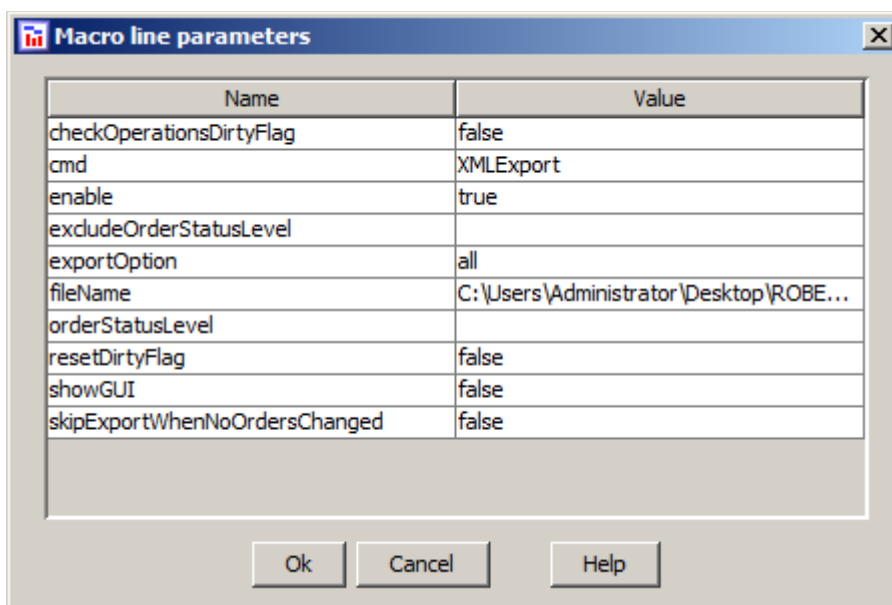


Kuva 13: Rob-Ex makroeditori

Makrotoiminto voidaan määrittää luomalla ensin makroiikkunan yläosassa olevaan kenttään uusi makrotoiminto ja sen jälkeen määrittää ikkunan alemmissä kentissä makrotoiminnon sisältämät tehtävät siirtämällä halutut tehtävät vasemmasta kentästä oikeaan ikkunaan sijoitettujen nuolinäppäimien avulla.

Makrotoiminto voidaan asettaa ajettavaksi tietyin väliajoin asettamalla aika ikkunan keskivaiheilla olevaan kenttään ja painamalla sen jälkeen Start -nappulaa. Makron teh-

tävän tarkemmat asetukset tulee määrittää tupla klikkaamalla ikkunan alalaidassa oikeanpuoleiseen kenttään siirrettyä tehtävää. Tässä tapauksessa tehtävänä on XML Export jonka asetukset tulee määrittää seuraavasti:



Kuva 14: XML Export -asetukset

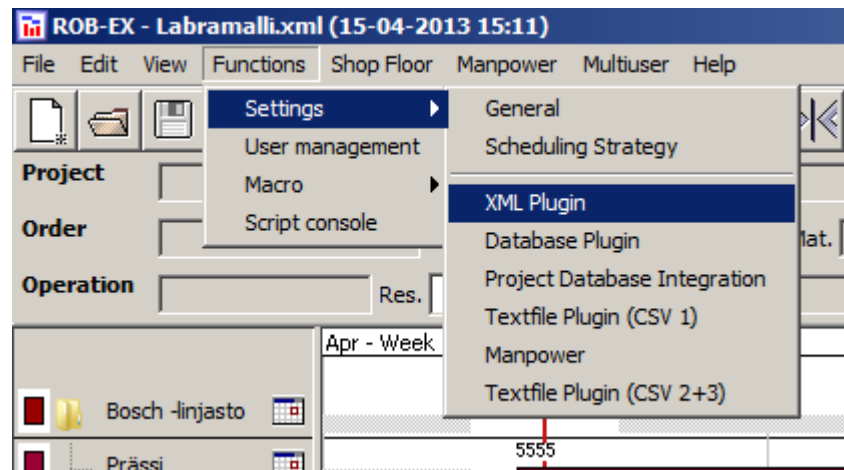
**Huom! Tiedostonimeksi täytyy määrittää haluttu Rob-Ex:n tiedostojen kansio tapauskohtaisesti. Kuvassa näkyvä tiedostonimi on vain esimerkkinä.**

## 9.2 Tietojen lukeminen

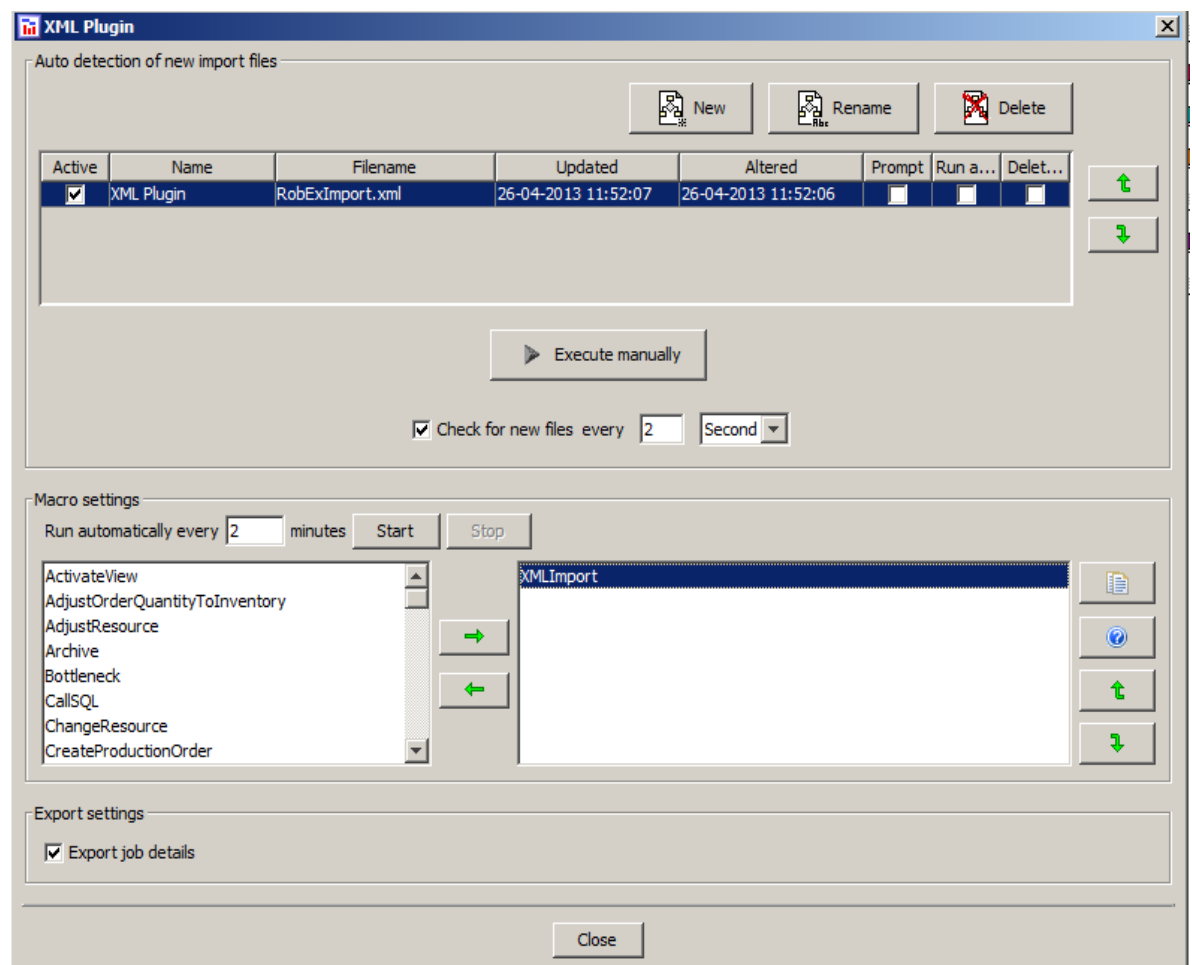
Luettaessa työkalun päivittämiä tietoja Rob-Ex:iin hyödynnetään Rob-Ex:n valmiita makrotoimintoja, joiden avulla voidaan määrittää päivitetystä tilaustiedoista koostuva XML -tiedosto luettavaksi halutusta kohdekansioista.

Makrotoimintona tietojen kirjoittamiseksi suositellaan käytettävän Rob-Ex:n valikosta valitsemalla "Functions→Settings→XML Plugin" löytyvää makroeditoria, johon voidaan määrittää XML Import -niminen makrotoiminto.

**Huom! XML Plugin on tarkoitettu tietojen päivittämiseksi ERP -järjestelmän kanssa. Makrotoimintoa hyödynnetään tässä koska se sisältää mahdollisuuden tiedoston päivittämisen tarkkailemiseen. XML Import- makro voidaan määrittää toimivaksi aikasyklillä myös Rob-Ex:n makroeditoriin.**



Kuva 15: XML Plugin valinta



Kuva 16: XML Plugin editori

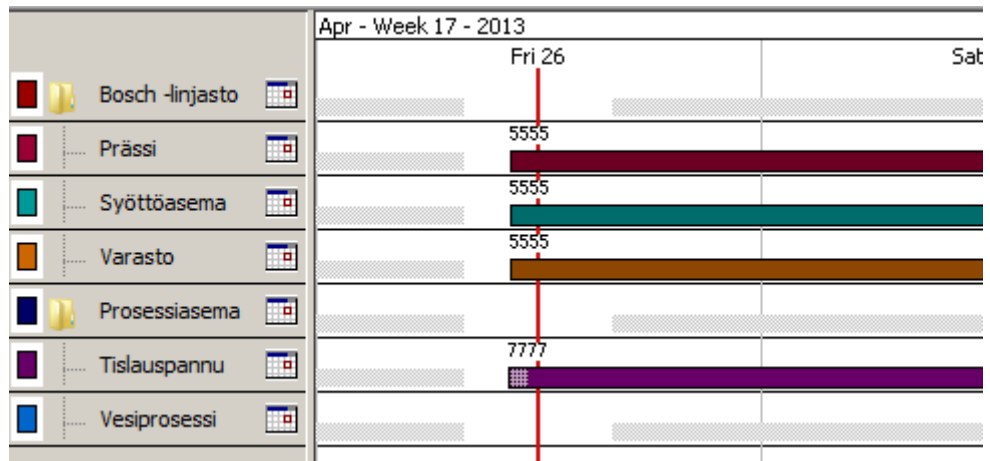
Makrotoiminto voidaan määrittää luomalla ensin makroikkunan yläosassa olevaan kenttään uusi makrotoiminto ja sen jälkeen määrittellä ikkunan alempiin kenttiin makrotoiminnon sisältämät tehtävät, siirtämällä halutut tehtävät vasemmasta kentästä oikeaan ikkunaan sijoitettujen nuolinäppäimien avulla. Tässä editorissa XML Import -tehtävä on valmiiksi valittuna jokaisessa makrotoiminnossa.

Luettavan tiedoston tiedostopolku määritetään ikkunan yläosaan määritetyn makrotoiminnon yhteyteen tupla klikkaamalla ja sen jälkeen tiedosto valitsemalla tai tiedostopolku kirjoittamalla. Makrotoiminto asetetaan ajettavaksi aina kun luettava tiedosto päivittyy valitsemalla ikkunan yläosaan määritelty makrotoiminto aktiiviseksi. Ikkunan keskellä olevaan kenttään asetetaan aikaväli, jolloin tarkistetaan onko luettavaksi määritelty tiedosto muuttunut. Mikäli tiedosto on päivittynyt, makrotoiminto lukee tiedot ja päivittää ne Rob-Ex:ään.

***Huom! Makrotoiminto voidaan myös asettaa ajettavaksi tietyin aikaväleihin mutta se ei ole suositeltavaa päivitysvälin ollessa lyhyehkö. Tällöin tietojen kirjoittamisen ja lukemisen välille jäävässä aikaikkunassa tehdyt muutokset saattavat hävitä, ts. Rob-Ex:ään päivitetään vanhoja tietoja tehtyjen muutosten päälle.***

### 9.3 Resurssiryhmät

Rob-Ex:ään on suositeltavaa määrittellä kutakin automaatiolaitteistoa tai koneryhmää kuvaava resurssiryhmä. Tällöin tiedonvälitystyökalu osaa tarvittaessa jakaa tilaukset oikeisiin tiedostoihin valvomoiden luettavaksi. Mikäli tilaustietoja yritetään kirjoittaa ja lukea useamman kuin yhden valvomon kanssa, eikä resurssiryhmiä ole Rob-Ex:ään määritelty, antaa työkalu virheilmoituksen.



Kuva 17: Resurssiryhmät

Resurssiryhmät voi määrittää Rob-Ex:n resurssieditorista. Kuvassa on määriteltynä esimerkkiryhmät, joissa laitteistot jaettuina linjastoon sekä prosessiasemaan. Lisätietoja aiheesta löytyy mm. Rob-Ex Gantt manual -käyttöohjeesta.

## ReadWrite for Rob-Ex v1.0 -tiedonvälityskalun ohjelmakoodi

'TÄRKEÄÄ KOMMENTTIA!

'Toiminta:

'1. Määritä käytettävien tiedostojen tiedostopolut.

'2. Lue tiedot Robexin xml -tiedostosta taulukoihin.

'3. Heitä tiedot fiksusti eteenpäin valvomolle. - CSV -tiedostoina ennalta määritetyssä formaatissa.

'4. Lue valvomon antamat tiedot ja päivitä taulukot.

'5. Päivitä taulukon tiedot xml -kirjoitustiedostoon Robexille.

'Ohjelma päivittää operaatioiden tilat tilausten statusten perusteella, tai valittaessa operaatiokohtaisesti hieman erilaisista tiedostoista.

'Roskakappaleet näkyvät myös joko tilauskohtaisesti jokaisessa operaatiossa yhteysuurina tai luetaan ja päivitetään operaatiokohtaisesti.

'Valmistuneet kappaleet päivitetään aina konekohtaisesti.

'Dataruutuun kirjoitetaan vain tilausten tiedot.

'Ohjelmalla ei ole tarkoitus lentää Marsiin.. :/

'ERROR HANDLERIT -> ON

'Importit:

Imports System.Xml

Imports System.IO

Imports System.Text

Public Class Form1

'Yleiset muuttujat ja taulukot:

Dim sTiedostopolkuReadRobex As String = ""

Dim sTiedostopolkuWriteRobex As String = ""

Dim TiedostopolutReadValvomo(0) As String 'Taulukoitu koska pitää pystyä luokemaan useita tiedostoja

Dim sTiedostopolkuWriteValvomo As String = ""

Dim iAikavali As Integer = 0

Dim i As Integer = 0

Dim n As Integer = 0

Dim k As Integer = 0

Dim p As Integer = 0

Dim a As Integer = 0

Dim q As Integer = 0

Dim t As Integer = 0



```
Dim TilausID() As String
Dim Aloitusaika() As String
Dim Aloitusaikaoperaatiot() As String
Dim Asiakas() As String
Dim Kpl() As String
Dim Tila() As String
Dim Tuote() As String
Dim OperaatioID() As String
Dim Operaatiot() As String
Dim Operaatiotila() As String
Dim Operaatioresurssinimi() As String
Dim Operaatioresurssiryhma() As String
Dim Resurssinimi() As String
Dim Resurssityyppi() As String
Dim Resurssiryhma() As String
Dim Valmistuneet() As String
Dim Hylatyt() As String
```

```
Dim bOnOff As Boolean = False
Dim bResurssitluettu As Boolean = False
Dim bResurssiryhmat As Boolean = False
Dim bKirjoitettavat_tilaukset As Boolean = False
Dim bTilauksetluettu As Boolean = False
Dim bOperaatiotluettu As Boolean = False
Dim bTiedostotkirjoitettu As Boolean = False
Dim bTiedostopolutOK As Boolean = False
Dim bxmlMuuttunut As Boolean = False
```

```
Dim bSetUpaikaloytyi As Boolean = False
Dim bcsvOperaatiotPilkku As Boolean = False
```

```
Dim xmlTarkkailija As New FileSystemWatcher 'Määritellään luettavaa .xml -
tiedostoa tarkkaileva toiminto.
```

'Määritely tänne koska pitää päästä vaikuttamaan toimintaan useammasta ali-ohjelmasta käsin.

'Määritellään koodausmuodot joita käytetään valvomolle kirjoitettavissa ja valvomolta luettavissa tiedostoissa.

```
Dim Koodausmuoto As Encoding
Dim KoodausmuotoUTF8 As New System.Text.UTF8Encoding(False) 'UTF-8 ei BOM:ia
Dim KoodausmuotoUTF8BOM As New System.Text.UTF8Encoding(True) 'UTF-8 ja BOM
Dim koodausmuotoUTF32 As New System.Text.UTF32Encoding 'UTF-32
Dim KoodausmuotoANSI As New System.Text.AsciiEncoding 'ANSI eli ASCII
Dim KoodausmuotoUNICODE As New System.Text.UnicodeEncoding 'UNICODE
```

'Määritely tänne koska pitää päästä vaikuttamaan toimintaan useammasta ali-ohjelmasta käsin.

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
'Määritetään valvomolta luettavien tiedostopolkujen taulukon ensimmäisen sarakkeen sisällöksi tyhjä merkkijono ohjelman toiminnallisista syistä johtuen.
```

```
TiedostopolutReadValvomo(0) = ""  
Label2.BackColor = Color.Black
```

```
End Sub
```

```
Sub OnOff()
```

```
Tiedostopolut()
```

```
'On/Off set-reset määrittäminen
```

```
If bOnOff = False And bTiedostopolutOK = True Then
```

```
    bOnOff = True  
    Label2.BackColor = Color.White  
    Label4.Text = "On"  
    CheckBox1.Enabled = False  
    CheckBox2.Enabled = False  
    CheckBox3.Enabled = False  
    CheckBox4.Enabled = False
```

```
    RadioButton4.Enabled = False  
    RadioButton5.Enabled = False  
    RadioButton6.Enabled = False  
    RadioButton7.Enabled = False  
    RadioButton8.Enabled = False
```

```
Else
```

```
    bOnOff = False  
    Label2.BackColor = Color.Black  
    Label4.Text = "Off"  
    CheckBox1.Enabled = True  
    CheckBox2.Enabled = True  
    CheckBox3.Enabled = True  
    If CheckBox3.Checked = True Then  
        CheckBox4.Enabled = True  
    End If
```

```
    RadioButton4.Enabled = True  
    RadioButton5.Enabled = True  
    RadioButton6.Enabled = True  
    RadioButton7.Enabled = True  
    RadioButton8.Enabled = True
```

```
End If
```

```
If bOnOff = True Then
```

```
    Tekstin_koodaus()  
    Tarkista_luettava_xml()
```

```
    Label10.Text = "Työskentelee"
```

```
    Luexmlresurssit()  
    Luexmltilaukset()  
    Luexmloperaatiot()  
    Kirjoitavalvomolle()  
    Luevalvomolta()  
    Taytataruutu()  
    Kirjoitaxml()
```

```
    bxmlMuuttunut = False  
    xmlTarkkailija.EnableRaisingEvents = False
```

```
    Label10.Text = "Valmis"
```

```
End If
```

```
End Sub
```

```
Sub Tiedostopolut()
```

```
'Tiedostopolkujen määrittäminen: määrittää Rob-Ex:n luettavan ja kirjoitettavan .xml-  
tiedoston sekä valvomolta luettavan ja valvomolle kirjoitettavan .csv-tiedoston  
sijainnit. ERROR HANDLER -> ON
```

```
Dim omaXml As XmlDocument = New XmlDocument()
```

```
Try
```

```
If bOnOff = False And sTiedostopolkuReadRobex = "" Or sTiedosto-  
polkuWriteRobex = "" Then
```

```
    'Rob-Ex:lta luettavan .xml-tiedoston tiedostopolku:
```

```
    OpenFD.InitialDirectory = "C:\"  
    OpenFD.Title = "Valitse luettava xml -tiedosto:"  
    OpenFD.Filter = ".xml -tiedosto|*.xml"  
    OpenFD.FileName = ""  
    OpenFD.Multiselect = True  
    OpenFD.ShowDialog()
```

```
sTiedostopolkuReadRobex = OpenFD.FileName
```

```
TextBox1.Text = sTiedostopolkuReadRobex
```

```
'Jos ei määritetty tiedostopolkua niin:
```

```
If sTiedostopolkuReadRobex = "" Then
```

```
    MsgBox("Kaikkia tiedostopolkuja ei ole määritetty! Ohjelma  
pysähtyy..", vbExclamation, "Tietoja puuttuu!")
```

```
        bOnOff = False
        Label2.BackColor = Color.Black
        Label10.Text = "Seis"
        Exit Sub
    End If

    'Rob-Ex:lle kirjoitettavan .xml-tiedoston tiedostopolku:

    SaveFD.InitialDirectory = "C:\"
    SaveFD.Title = "Luo kirjoitettava xml -tiedosto:"
    SaveFD.Filter = ".xml -tiedosto|*.xml"
    SaveFD.FileName = ""
    SaveFD.ShowDialog()

    sTiedostopolkuWriteRobex = SaveFD.FileName

    TextBox2.Text = sTiedostopolkuWriteRobex

    'Jos ei määritetty tiedostopolkua niin:

    If sTiedostopolkuWriteRobex = "" Then

        MsgBox("Kaikkia tiedostopolkuja ei ole määritetty! Ohjelma
        pysähtyy..", vbExclamation, "Tietoja puuttuu!")
        bOnOff = False
        Label2.BackColor = Color.Black
        Label10.Text = "Seis"
        Exit Sub

    Else

        'Xml -tiedoston avaaminen ja kopiointi kohdekansioon

        omaXml.Load(sTiedostopolkuReadRobex)

        omaXml.Save(sTiedostopolkuWriteRobex)

        MsgBox("Rob-Ex:n luettava tiedosto valittu ja kirjoitettava
        tiedosto luotu onnistuneesti.", vbInformation, "Tiedostot va-
        littu")
    End If
End If

'Valvomolta luettavan .csv-tiedoston tiedostopolku:

If bOnOff = False And TiedostopolutReadValvomo(0) = "" Or sTiedosto-
polkuWriteValvomo = "" Then

    t = 0

    If CheckBox1.Checked = True Then

        OpenFD.Title = "Valitse luettavat valvomoiden .csv-
        tiedostot:"
        OpenFD.Multiselect = True
```

Else

```
OpenFD.Title = "Valitse luettava valvomon .csv-tiedosto:"  
OpenFD.Multiselect = False
```

End If

```
OpenFD.InitialDirectory = "C:\"  
OpenFD.Filter = ".csv -tiedosto|*.csv"  
OpenFD.FileName = ""  
OpenFD.ShowDialog()
```

For Each filename In OpenFD.FileNames

```
ReDim Preserve TiedostopolutReadValvomo(t)  
TiedostopolutReadValvomo(t) = OpenFD.FileNames(t)  
t = t + 1
```

Next

```
TextBox3.Text = TiedostopolutReadValvomo(0) + "(useita  
tiedostoja)"
```

'Jos ei määritetty tiedostopolkua niin:

If TiedostopolutReadValvomo(0) = "" Then

```
MsgBox("Kaikkia tiedostopolkuja ei ole määritetty! Ohjelma  
pysähtyy..", vbExclamation, "Tietoja puuttuu!")  
bOnOff = False  
Label2.BackColor = Color.Black  
Label10.Text = "Seis"  
Exit Sub
```

End If

'Valvomolle kirjoitettavan .csv-tiedoston tiedostopolku:

```
SaveFD.InitialDirectory = "C:\"
```

'Tarkistetaan kirjoitetaanko yksi vai useampia tiedostoja

If CheckBox1.Checked = True Then

```
SaveFD.Title = "Luo valvomoille kirjoitettavat .csv-tiedostot  
- yksi tiedostonimi ilman päätettä:"  
SaveFD.Filter = ""
```

Else

```
SaveFD.Title = "Luo kirjoitettava valvomon .csv-tiedosto:"  
SaveFD.Filter = ".csv -tiedosto|*.csv"
```

End If

```
SaveFD.FileName = ""  
SaveFD.ShowDialog()
```

```
sTiedostopolkuWriteValvomo = SaveFD.FileName
```

```
    If CheckBox1.Checked Then
        TextBox4.Text = sTiedostopolkuWriteValvomo + "(useita tiedos-
toja)"
    Else
        TextBox4.Text = sTiedostopolkuWriteValvomo
    End If

    'Jos ei määritetty tiedostopolkua niin:

    If sTiedostopolkuWriteValvomo = "" Then

        MsgBox("Kaikkia tiedostopolkuja ei ole määritetty! Ohjelma
pysähtyy..", vbExclamation, "Tietoja puuttuu!")
        bOnOff = False
        Label2.BackColor = Color.Black
        Label10.Text = "Seis"
        Exit Sub

    Else

        MsgBox("Valvomon luettava tiedosto valittu ja kirjoitettava
tiedosto luotu onnistuneesti.", vbInformation, "Tiedostot va-
littu")
        bTiedostopolutOK = True
    End If
End If

Catch ex As Exception

    MsgBox("Tiedostojen määrittäminen ei onnistunut!", vbCritical, "Virhe!")

    bOnOff = False
    Label2.BackColor = Color.Black
    Label10.Text = "Seis"

End Try

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles Timer1.Tick

    'Timer -loop jonka avulla aliohjelmat suoritetaan valitulla aikavälillä
tietojen päivittämisen takaamiseksi

    If bOnOff = True Then

        If CheckBox5.Checked = True Then 'Jos päivitetään luettavan tiedoston
muuttuessa/päivittyessä niin näin:

            Tarkista_luettava_xml()
            Tekstin_koodaus()

            If bxmlMuuttunut = True Then
                Label10.Text = "Työskentelee"
```

```
        Luexmlresurssit()
        Luexmltilaukset()
        Luexmloperaatiot()
        Kirjoitavalvomolle()
        Luevalvomolta()
        Taytataruutu()
        Kirjoitaxml()

        bxmlMuuttunut = False
        xmlTarkkailija.EnableRaisingEvents = False

        Label10.Text = "Valmis"
    End If

Else 'Jos taas normaalilla aikasyklillä niin näin:

    Tarkista_luettava_xml()
    Tekstin_koodaus()

    Label10.Text = "Työskentelee"

    Luexmlresurssit()
    Luexmltilaukset()
    Luexmloperaatiot()
    Kirjoitavalvomolle()
    Luevalvomolta()
    Taytataruutu()
    Kirjoitaxml()

    Label10.Text = "Valmis"

    bxmlMuuttunut = False
    xmlTarkkailija.EnableRaisingEvents = False

End If
End If
End Sub
Sub Tarkista_luettava_xml()

'Tarkistaa onko luettava .xml -tiedosto päivittynyt entisestä kohdekansi-
ossa. - ERROR HANDLER -> ON

Try

    xmlTarkkailija.Path = Path.GetDirectoryName(sTiedostopolkuReadRobex)
    xmlTarkkailija.NotifyFilter = (NotifyFilters.LastWrite)

    AddHandler xmlTarkkailija.Changed, AddressOf Muuttunutxml
    xmlTarkkailija.EnableRaisingEvents = True
Catch ex As Exception

    MsgBox("Luettavan .xml -tiedoston tarkkailu ei onnistunut!", vbCritic-
al, "Virhe!")
```

```
        bOnOff = False
        Label2.BackColor = Color.Black
        Label10.Text = "Seis"
        Exit Sub

    End Try

End Sub
Private Sub Muuttunutxml(ByVal source As Object, ByVal e As FileSystemEventArgs)

    'Kun havaitaan että tiedosto on muuttunut muutetaan muuttujan arvoa ;)

    bxmlMuuttunut = True

End Sub

Sub Tekstin_koodaus()

    'Määrittää mitä koodausta käytetään valvomon kirjoitettavissa ja luettavissa csv -tiedostoissa:

    If RadioButton4.Checked = True Then
        Koodausmuoto = KoodausmuotoANSI
    End If

    If RadioButton5.Checked = True Then
        Koodausmuoto = KoodausmuotoUNICODE
    End If

    If RadioButton6.Checked = True Then
        Koodausmuoto = KoodausmuotoUTF8
    End If

    If RadioButton7.Checked = True Then
        Koodausmuoto = KoodausmuotoUTF8BOM
    End If

    If RadioButton8.Checked = True Then
        Koodausmuoto = koodausmuotoUTF32
    End If

End Sub

Sub Luexmlresurssit()

    'Lukee resurssit, niiden resurssiryhmät ja tyypit - ERROR HANDLER -> ON

    If bOnOff = True Then

        Try

            a = 0

            ReDim Resurssinimi(a)
```



```
ReDim Resurssiryhma(a)
ReDim Resurssityyppi(a)

Using xmlresurssilukija As XmlReader = XmlReader.
Create(sTiedostopolkuReadRobex)

While xmlresurssilukija.Read()

    If xmlresurssilukija.IsStartElement() Then

        If xmlresurssilukija.Name = "Resource" Then

            Dim sResurssinimi As String = xmlresurssiluki-
            ja("name")

            ReDim Preserve Resurssinimi(a)

            Dim sResurssityyppi As String = xmlresurssiluki-
            ja("type")
            ReDim Preserve Resurssityyppi(a)

            'Tarkistetaan onko kyseessä yksittäinen resurssi
            vai ryhmä xml -tiedoston erinimisten id määritys-
            ten vuoksi

            Dim sResurssiryhma As String

            If sResurssityyppi = "group" Then

                sResurssiryhma = xmlresurssilukija("id")
                ReDim Preserve Resurssiryhma(a)

            Else

                sResurssiryhma = xmlresurssiluki-
                ja("resGroupId")

                ReDim Preserve Resurssiryhma(a)

            End If

            'Tietojen kirjoitus taulukoihin

            Resurssinimi(a) = sResurssinimi
            Resurssityyppi(a) = sResurssityyppi
            Resurssiryhma(a) = sResurssiryhma

            a = a + 1

        End If
    End If
End While
End Using

If CheckBox1.Checked = True Then
```

```
Tarkistaressit() 'Tarkistetaan onko resurssit jaettu ryh-
miin siltä varalta jos halutaan kirjoittaa useampi tiedosto

End If

Catch ex As Exception

'Tiedosto ehkä varattu joten statusmuutos

Label10.Text = "Odottaa..."
End Try

bResurssitluettu = True
End If

End Sub

Sub Tarkistaressit()

Dim a2 As Integer

a2 = 0

For a2 = 0 To a - 1

If Resurssityyppi(a2) = "group" Then

bResurssiryhmat = True
Exit For

Else

bResurssiryhmat = False

End If

Next

If bResurssiryhmat = True Then

Exit Sub

Else

MsgBox("Resurssiryhmät puuttuvat Rob-Ex:sta joten useampia tiedostoja
ei voida kirjoittaa niiden perusteella. Ohjelma pysähtyy..", vbExcla-
mation, "Resurssiryhmät puuttuvat!")

bOnOff = False
Label4.Text = "Off"
Label2.BackColor = Color.Black
Label10.Text = "Seis"

End If
```

End Sub

Sub Luexmltilaukset()

'Xml -tiedoston lukeminen ja tilaustietojen taulukointi - ERROR HANDLER -  
> ON

If bOnOff = True And bResurssitluettu = True Then

Try

i = 0

ReDim TilausID(i)  
ReDim Aloitusaika(i)  
ReDim Asiakas(i)  
ReDim Kpl(i)  
ReDim Tila(i)  
ReDim Tuote(i)

Using xmltilauslukija As XmlReader = XmlReader.  
Create(sTiedostopolkuReadRobex)  
While xmltilauslukija.Read()

If xmltilauslukija.IsStartElement() Then

If xmltilauslukija.Name = "ProdOrder" Then

Dim sTilausnro As String = xmltilauslukija("name")

ReDim Preserve TilausID(i)

Dim sAloituspvm As String = xmltilauslukija("startCal")

ReDim Preserve Aloitusaika(i)

Dim sAsiakasnimi As String = xmltilauslukija("cust")

ReDim Preserve Asiakas(i)

Dim sKpl As String = xmltilauslukija("quantity")  
ReDim Preserve Kpl(i)

Dim sTila As String = xmltilauslukija("state")  
ReDim Preserve Tila(i)

```
'Tietojen kirjoitus taulukoihin

TilausID(i) = sTilausno
Aloitusaika(i) = sAloituspvm
Asiakas(i) = sAsiakasnimi
Kpl(i) = sKpl
Tila(i) = sTila

End If

If xmltilauslukija.Name = "Product" Then

    Dim sTuote As String = xmltilauslukija("name")
    'Tämä muutettu "item":stä

    ReDim Preserve Tuote(i)
    Tuote(i) = sTuote
    i = i + 1

End If

End If
End While
End Using

bTilauksetluettu = True
bResurssitluettu = False

Catch ex As Exception

    'Tiedosto ehkä varattu joten statusmuutos

    Label10.Text = "Odottaa..."

End Try
End If
End Sub

Sub Luexmloperaatiot()

    'Xml -tiedoston lukeminen ja operaatioiden taulukointi, taulukointi ta-
    pahtuu eri xml elementeistä joten hiukan sekavaa ERROR HANDLER -> ON

    If bOnOff = True And bTilauksetluettu = True Then

        Try

            n = 0
            k = 0

            ReDim OperaatioID(n)
            ReDim Operaatiot(n)
```

```
ReDim Operaatiotila(n)
ReDim Valmistuneet(n)
ReDim Operaatioresurssinimi(n)
ReDim Aloitusaikaoperaatiot(n)

Dim j As Integer

Using xmloperaatiolukija As XmlReader = XmlReader.Create(sTiedostopolkuReadRobex)
    While xmloperaatiolukija.Read()

        If xmloperaatiolukija.IsStartElement() Then

            If xmloperaatiolukija.Name = "ProdOrder" Then

                k = k + 1

            End If

            If xmloperaatiolukija.Name = "Opr" Then

                'Boolean muuttujan nollaus asetustajan etsimistä
                koskien

                bSetUpaikaloytyi = False

                'Tietojen luku tiedostosta

                Dim sOperaatio As String = xmloperaatiolukija("name")

                ReDim Preserve Operaatiot(n)

                Dim sOperaatiotila As String = xmloperaatiolukija("state")

                ReDim Preserve Operaatiotila(n)

                'Nämä kolme taulukkoa määritetään oikean kokoiseksi
                ReDim Preserve OperaatioID(n)
                ReDim Preserve Valmistuneet(n)
                ReDim Preserve Hylatyt(n)

                'Tietojen kirjoitus taulukoihin

                Operaatiot(n) = sOperaatio
                Operaatiotila(n) = sOperaatiotila
                OperaatioID(n) = TilausID(k - 1)

            End If

            'Resurssiryhmän liittäminen operaatioihin
```

```
If xmloperaatiolukija.Name = "ResAlloc" Then
    ReDim Preserve Operaatioresurssinimi(n)

    Dim sResurssinimioperaatiolukija As String =
        xmloperaatiolukija("resName")

    Operaatioresurssinimi(n) = sResurssinimioperaatiolukija

    For j = 0 To a - 1

        If Resurssinimi(j) = sResurssinimioperaatiolukija Then

            ReDim Preserve Operaatioresurssiryhma(n)

            Operaatioresurssiryhma(n) = Resurssiryhma(j)

        End If

    Next

End If

'Operaation alkamisajankohdan lisääminen:
If xmloperaatiolukija.Name = "JobEntry" Then

    ReDim Preserve Aloitusaikaoperaatiot(n)

    Dim sAloitusaikaoperaatio As String = xmloperaa
        tiolukija("startCal")

    'Asetusajan löytyessä operaatiosta laitetaan boolean muuttuja todeksi että saadaan oikea aika luettua talteen

    If xmloperaatiolukija("name") = "setUp" Then

        bSetUpaikaloityi = True

    End If

    'Suodatetaan joko asetusaika tai työn aloitusaika

        If xmloperaatiolukija("name") = "setUp" And
            bSetUpaikaloityi = True Then

            Aloitusaikaoperaatiot(n) = sAloitusaikaoperaatio
            n = n + 1

        ElseIf xmloperaatiolukija("name") = "w1" And
            bSetUpaikaloityi = False Then
```

```
Aloitusaikaoperaatiot(n) = sAloitusaikaope-
raatio
n = n + 1

End If
End If
End If
End While
End Using

bTilauksetluettu = False
bOperaatiotluettu = True

Catch ex As Exception

'Tiedosto ehkä varattu joten statusmuutos

Label10.Text = "Odottaa..."

End Try
End If
End Sub

Sub Kirjoitavalvomolle()

'Valitaan kirjoitetaanko yksi vai monta tiedostoa eri resurssiryhmille -
ERROR HANDLER -> NaN

If CheckBox1.Checked = True Then
Kirjoitamontatiedostoa()
Else
Kirjoitayksitiedosto()
End If
End Sub

Sub Kirjoitamontatiedostoa()

'Kirjoittaa tilaukset useaan eri tiedostoon resurssiryhmien perusteella -
ERROR HANDLER -> ON

Dim v As Integer
Dim w As Integer
Dim s As Integer
Dim r As Integer

Dim ValvomWrite As IO.StreamWriter

v = 0
s = 0
w = 0
r = 0

If bOnOff = True And bOperaatiotluettu = True Then
```

Try

'Etsitään resurssiryhmät taulukosta ja toimitaan niiden perusteella:

```
For v = 0 To a - 1
```

```
    If Resurssityyppi(v) = "group" Then
```

```
        'Tiedostoon kirjoitus
```

```
ValvomWrite =
```

```
My.Computer.FileSystem.OpenTextFileWriter(sTiedostopolkuWriteValvomo + " " +  
Resurssinimi(v) + ".csv", False, Koodausmuoto)
```

```
    For s = 0 To i - 1
```

```
        'Kirjoitetaan valvomolle vain 'Start Possible',  
        'Started' tai 'Paused' -tiloissa olevat tilaukset
```

```
        If Tila(s) = 40 Or Tila(s) = 50 Or Tila(s) = 60 Then
```

```
            For w = 0 To n - 1
```

```
                If Resurssiryhma(v) = Operaatioresurssiryhma(w) And OperaatioID(w) = TilausID(s)  
                Then
```

```
                    If CheckBox2.Checked = True Then
```

```
                        'Operaatioiden kirjoitus tilausten alle:
```

```
                            ValvomWrite.WriteLine(TilausID(s) + "," + Aloitusaika(s) + "," + Asiakas(s) +  
                            "," + Tuote(s) + "," + Kpl(s))
```

```
                                For r = 0 To n - 1
```

```
                                    If OperaatioID(r) = TilausID(s) And Operaatioresurssiryhma(r) = Resurssiryhma(v)  
                                    Then
```

```
                                        'Tarkistetaan laitetaanko pilkkua operaation eteen jos operaatio kirjoitetaan  
                                        toisen perään
```

```
                                            If bcsvOperaatiotPilkku = True Then
```

```
                                                ValvomWrite.Write("," + Aloitusaikaoperatiot(r) + "," + Operaatiot(r) + "," +  
                                                Operaatioresurssinimi(r))
```

```
                                            ElseIf bcsvOperaatiotPilkku = False Then
```

```
                                                ValvomWrite.Write(Aloitusaikaoperatiot(r) + "," + Operaatiot(r) + "," + Operaa-  
                                                tioresurssinimi(r))
```

```
                                            bcsvOperaatiotPilkku = True
```

```
                                        End If
```

```
                                    End If
```

```
                                Next
```

```
                    ValvomWrite.WriteLine("")
```



```

bcsvOperaatiotPilkku = False
Exit For

Else

'Pelkkien tilausten kirjoitus

ValvomWrite.WriteLine(TilausID(s) + "," + Aloitus aika(s) + "," + Asiakas(s) +
"," + Tuote(s) + "," + Kpl(s))

Exit For
End If
End If
Next
End If
Next
ValvomWrite.Close()
End If
Next

Catch ex As Exception
'Tiedosto ehkä varattu joten statusmuutos:
Label10.Text = "Odottaa.."
End Try
End If

bKirjoitettavat_tilaukset = False
bOperaatiotluettu = False
bTiedostotkirjoitettu = True

End Sub

Sub Kirjoitayksitiedosto()

'Kirjoittaa tilaustiedot valvomolle .csv -tiedostona - ERROR HANDLER -> ON

Try

If bOnOff = True And bOperaatiotluettu = True Then

Dim g As Integer = 0
Dim v As Integer = 0

If bOnOff = True Then
g = 0
v = 0

Dim ValvomWrite As IO.StreamWriter =
My.Computer.FileSystem.OpenTextFileWriter(sTiedostopolkuWriteValvomo, False,
Koodausmuoto)

For g = 0 To i - 1

'Kirjoitetaan valvomolle vain 'Start Possible', 'Started' tai 'Paused' -tiloissa
olevat tilaukset

```

```
If Tila(g) = 40 Or Tila(g) = 50 Or Tila(g) = 60 Then

'Operaatioiden kirjoitus tilausten alle

    If CheckBox2.Checked = True Then
    ValvomWrite.WriteLine(TilausID(g) + "," + Aloitusaika(g) + "," + Asiakas(g) +
    "," + Tuote(g) + "," + Kpl(g))

        For v = 0 To n - 1

            If OperaatioID(v) = TilausID(g) The

' Tarkistetaan laitetaanko pilkkua operaation eteen jos operaatio kirjoitetaan
toisen perään

                If bcsvOperaatiotPilkku = True Then
                ValvomWrite.Write("," + Aloitusaikaoperaatiot(v) + "," + Operaatiot(v) + "," +
                Operaatioresurssinimi(v))

                    Else

                        ValvomWrite.Write(Aloitusaikaoperaatiot(v) + "," + Operaatiot(v) + "," + Ope-
                        raatioresurssinimi(v))
                        bcsvOperaatiotPilkku = True

                    End If
                End If
            Next

                ValvomWrite.WriteLine("")
                bcsvOperaatiotPilkku = False

            Else

'Pelkkien tilausten kirjoitus

                ValvomWrite.WriteLine(TilausID(g) + "," + Aloitusaika(g) + "," + Asiakas(g) +
                "," + Tuote(g) + "," + Kpl(g))

                    End If
                End If
            Next

                ValvomWrite.Close()

            End If
        End If

Catch ex As Exception

    'Tiedosto ehkä varattu joten statusmuutos:
    Label10.Text = "Odottaa..."

    End Try
    bOperaatiotluettu = False
    bTiedostotkirjoitettu = True
End Sub
```

```
Sub Luevalvomolta()
```

```
'Lukee tilat ja valmistuneet kappaleet valvomon kirjoittamasta .csv tiedostosta  
ja päivittää taulukoiden datan ERROR HANDLER - ON
```

```
Dim b As Integer = 0
```

```
Try
```

```
  If bOnOff = True And bTiedostotkirjoitettu = True Then
```

```
    For b = 0 To t - 1
```

```
      Using Valvomoread As New Microsoft.VisualBasic.FileIO.  
        TextFieldParser(TiedostopolutReadValvomo(b))
```

```
        Valvomoread.TextFieldType =  
          Microsoft.VisualBasic.FileIO.FieldType.Delimited  
        Valvomoread.SetDelimiters(",")
```

```
        Dim sLuettudata As String()  
        Dim sLuetutoperaatiot As String()  
        Dim m As Integer = 0  
        Dim d As Integer = 0  
        Dim o As Integer = 0  
        Dim p As Integer = 0  
        Dim e As Integer = 0
```

```
      While Not Valvomoread.EndOfData
```

```
        ReDim sLuettudata(0)  
        ReDim sLuetutoperaatiot(0)
```

```
'Luetaan ensin tilausrivi ja mikäli yksityiskohtaisempi  
päivitysoptio on valittu operaatorivi sen jälkeen.
```

```
        sLuettudata = Valvomoread.ReadFields()  
        Dim iDatanpituus As Integer = sLuettudata.GetLength(0)
```

```
        If CheckBox3.Checked Then  
          sLuetutoperaatiot = Valvomoread.ReadFields()  
        End If
```

```
        m = 0  
        d = 0  
        o = 0  
        p = 0  
        e = 0
```

```
      For m = 0 To i - 1 'Tilauslista läpi
```

```
        If sLuettudata(0) = TilausID(m) Then
```

```
          Tila(m) = sLuettudata(1) 'Päivitetään tilausten tilat
```

```
For o = 0 To n - 1 'Operaatiolista läpi

'Tarkistetaan onko valittuna tila- ja kappalemääräpäivitykset operaatiokohtaisesti

If CheckBox3.Checked = True Then

Dim iDatanpituusoperaatiot As Integer = sLuetutoperaatiot.GetLength(0)

If OperaatioID(o) = TilausID(m) Then 'Tarkistetaan tilausnumeroiden vastaavuus operaatiolistalla

Do Until p >= iDatanpituusoperaatiot 'Koska tiedot peräkkäin tiedostossa niin loopataan läpi kunnes loppuu

For e = 0 To n - 1 'Käydään taas taulukkoa läpi että saadaan oikeisiin kohtiin päivitettyä tietoja

'Tarkistetaan että Operaation nimi, Resurssin nimi ja ID -numero täsmäävät:

If Operaatioresurssinimi(e) = sLuetutoperaatiot(p + 1) And OperaatioID(e) = sLuettudata(0) And Operaatiot(e) = sLuetutoperaatiot(p) Then

Operaatiotila(e) = sLuetutoperaatiot(p + 2)

Valmistuneet(e) = sLuetutoperaatiot(p + 3)

'Katsotaan päivitetäänkö hukkatavara operaatiosta vai tilauksesta:

If CheckBox4.Checked = True Then

Hylatyt(e) = sLuetutoperaatiot(p + 4)

Else

Hylatyt(e) = sLuettudata(2)

End If

End If

Next
```

```
'Kasvatetaan indeksiä vasta loopin jälkeen ettei pääse liian suureksi
If CheckBox4.Checked = True Then
    p = p + 5
Else
    p = p + 4
End If

Loop
End If

Else 'Tavallinen päivitys jossa tilatiedot päivitetään tilauksen
mukaan

'Operaatioiden tilatietojen päivittäminen vastaamaan kunkin tilauksen
tilatietoja

If OperaatioID(o) = TilausID(m) Then 'Tarkistetaan tilausnumeroiden
vastaavuus operaatiolistalla
Hylatyt(o) = sLuettudata(2)

    'Vaihdetaan operaatioiden tilat vastaamaan tilausten tiloja

If Tila(m) = "50" Then 'Jos tilaus on tilassa 'Started' niin vaihdetaan
operaatio tilaan 'Started' ja merkitään päivitetyksi.
Operaatiotila(o) = "40"

ElseIf Tila(m) = "60" Then 'Jos tilaus on tilassa 'Paused' niin
vaihdetaan operaatio tilaan 'Paused' ja merkitään päivitetyksi.
Operaatiotila(o) = "50"

ElseIf Tila(m) = "80" Then 'Jos tilaus on tilassa 'Completed' niin
vaihdetaan operaatio tilaan 'Completed' ja merkitään päivitetyksi.
Operaatiotila(o) = "70"

Else
Operaatiotila(o) = "00"
End If

If (d + 3) < iDatanpituus Then

If Valmistuneet(o) <= sLuettudata(d + 3) Then
Valmistuneet(o) = sLuettudata(d + 3)
    End If
    End If
    d = d + 1
End If

End If
Next
End If
Next
End While
```

```
Valvomoread.Close()  
End Using  
Next  
End If
```

```
        bTiedostotkirjoitettu = False  
  
        Catch ex As Exception  
  
            'Tiedosto ehkä varattu joten statusmuutos:  
  
            Label10.Text = "Odottaa..."  
        End Try  
  
End Sub
```

```
Sub Taytadataruutu()  
  
    'Laittaa tilaustiedot dataruudukkoon - ERROR HANDLER - ON  
  
    Try  
  
        If bOnOff = True Then  
  
            DataGridView1.Rows.Clear()  
  
            'Ettei tule nollaerroria  
  
            If i > 1 Then  
                DataGridView1.Rows.Add(i - 1)  
            End If  
  
            For z = 0 To i - 1  
  
                'Merkitään viivalla jos xml -tiedostosta luetun paramet-  
                rin arvo sattuu olemaan null  
  
                If TilausID(z) = "" Then  
                    DataGridView1.Rows.Item(z).Cells(0).Value = "-"  
                Else  
                    DataGridView1.Rows.Item(z).Cells(0).Value = TilausID(z)  
                End If  
  
                If Aloitusaika(z) = "" Then  
                    DataGridView1.Rows.Item(z).Cells(1).Value = "-"  
                Else  
                    DataGridView1.Rows.Item(z).Cells(1).Value = Aloitusaika(z)  
                End If  
  
                If Asiakas(z) = "" Then  
                    DataGridView1.Rows.Item(z).Cells(2).Value = "-"  
                End If  
  
            Next z  
  
        End Try  
  
End Sub
```

```
Else
    DataGridView1.Rows.Item(z).Cells(2).Value = Asiakas(z)
End If

If Kpl(z) = "" Then
    DataGridView1.Rows.Item(z).Cells(3).Value = "-"
Else
    DataGridView1.Rows.Item(z).Cells(3).Value = Kpl(z)
End If

If Tuote(z) = "" Then
    DataGridView1.Rows.Item(z).Cells(4).Value = "-"
Else
    DataGridView1.Rows.Item(z).Cells(4).Value = Tuote(z)
End If
Next
End If

Catch ex As Exception

    MsgBox("Tietojen päivittäminen ruudukkoon ei onnistunut!", vbCritical, "Virhe!")
    bOnOff = False
    Label4.Text = "Off"
    Label2.BackColor = Color.Black
    Label10.Text = "Seis"

End Try

End Sub

Sub KirjoitaXml()

    'Xml -tiedoston kirjoittaminen ja tietojen päivitys taulukon perusteella
    - ERROR HANDLER -> ON

    Try

        If bOnOff = True Then

            Dim OmaXmlW As XmlDocument = New XmlDocument()

            OmaXmlW.Load(sTiedostopolkuReadRobex)

            Dim OmaXmlWProgressTag As XmlElement =
            OmaXmlW.CreateElement("Progress")

            OmaXmlWProgressTag.SetAttribute("accQuantity", "0")
            OmaXmlWProgressTag.SetAttribute("accScrapQuantity", "0")

            Dim OmaXmlWRoot As XmlNode

            OmaXmlWRoot = OmaXmlW.DocumentElement

            'Käydään xml -tiedosto läpi ja päivitetään tiedot sekä lisätään
            <Progress> -tagit ja päivitetään nekin

            p = 0
```

```
q = 0
For Each OmaXmlWElement In OmaXmlWRoot.ChildNodes
    For Each OmaXmlWChild In OmaXmlWElement.Childnodes

        If OmaXmlWChild.Name = "ProdOrder" Then

            OmaXmlWChild.Attributes("state").Value = Tila(p)
            p = p + 1
        End If

        For Each OmaXmlWChildChild In OmaXmlWChild.Childnodes
            For Each OmaXmlWChildChildChild In OmaXmlWChild-
                Child.Childnodes

                If OmaXmlWChildChildChild.Name = "Opr" Then

                    OmaXmlWChildChild-
                    Child.Attributes("state").Value = Operaatitila(q)

                For Each OmaXmlWChildChildChildChild In OmaXmlWChildChild-
                    Child.Childnodes

                    If OmaXmlWChildChildChildChild.Name = "ResAllocList" And (Tila(p -
                    1) = "50" Or Tila(p - 1) = "60" Or Tila(p - 1) = "80") Then

                        OmaXmlWChildChild-
                        Child.InsertAfter(OmaXmlWProgressTag.Clone(), OmaXmlWChildChild-
                        ChildChild)

                    End If

                End If

            End If

            OmaXmlWChildChildChildChild.Name = "Progress" Then
                OmaXmlWChildChildChildChild-
                Child.Attributes("accQuantity").Value = Valmistuneet(q)
                OmaXmlWChildChildChildChild-
                Child.Attributes("accScrapQuantity").Value = Hylatyt(q)

            End If

        Next
        q = q + 1
    End If
```



```

                                Next
                            Next
                        Next
                    Next

                OmaXmlW.Save(sTiedostopolkuWriteRobex)

            End If

        Catch ex As Exception

            'Tiedosto ehkä varattu joten statusmuutos:
            Label10.Text = "Odottaa..."

        End Try

    End Sub

Sub Paivitatiedot()

    'Manuaalinen tietojen päivitys ja ohjelman ajo - ERROR HANDLER -> NaN

    Timer1.Stop()

    Tiedostopolut()

    If bTiedostopolutOK = True Then

        bOnOff = True

        CheckBox1.Enabled = False
        CheckBox2.Enabled = False
        CheckBox3.Enabled = False
        CheckBox4.Enabled = False

        RadioButton4.Enabled = False
        RadioButton5.Enabled = False
        RadioButton6.Enabled = False
        RadioButton7.Enabled = False
        RadioButton8.Enabled = False

    End If

    Tarkista_luettava_xml()
    Tekstin_koodaus()

    Label10.Text = "Työskentelee"

    Luexmlresurssit()
    Luexmltilaukset()
    Luexmloperaatiot()
    Kirjoitavalvomolle()
    Luevalvomolta()

```

```
Taytadataruutu()
Kirjoitaxml()

bxmlMuuttunut = False
xmlTarkkailija.EnableRaisingEvents = False

Label10.Text = "Valmis"

bOnOff = False
CheckBox1.Enabled = True
CheckBox2.Enabled = True
CheckBox3.Enabled = True
If CheckBox3.Checked = True Then
    CheckBox4.Enabled = True
End If

RadioButton4.Enabled = True
RadioButton5.Enabled = True
RadioButton6.Enabled = True
RadioButton7.Enabled = True
RadioButton8.Enabled = True

Label2.BackColor = Color.Black
Label4.Text = "Off"

Timer1.Start()

End Sub

Sub Vaihdattiedostopolut()

'Tiedostopolkujen vaihto ja ohjelman pysäytys mikäli käynnissä - ERROR
HANDLER -> NaN

If bOnOff = True Then
    MsgBox("Ohjelma pysähtyy koska tiedostopolut vaihdetaan.", vbInformation, "Tiedostopolkujen vaihto")
End If

bOnOff = False
Label4.Text = "Off"
Label2.BackColor = Color.Black
Label10.Text = "Seis"
sTiedostopolkuReadRobex = ""
sTiedostopolkuWriteRobex = ""
ReDim TiedostopolutReadValvomo(0)
TiedostopolutReadValvomo(0) = ""
sTiedostopolkuWriteValvomo = ""

Tiedostopolut()

End Sub

Sub VaihdattiedostopolutRobex()

If bOnOff = True Then
    MsgBox("Ohjelma pysähtyy koska tiedostopolut vaihdetaan.", vbInformation, "Tiedostopolkujen vaihto")
End If

```

```
End If

bOnOff = False
Label14.Text = "Off"
Label2.BackColor = Color.Black
Label10.Text = "Seis"
sTiedostopolkuReadRobex = ""
TextBox1.Clear()
sTiedostopolkuWriteRobex = ""
TextBox2.Clear()

If TiedostopolutReadValvomo(0) = "" Or sTiedostopolkuWriteValvomo = ""
Then
    MsgBox("Myös valvomon tiedostojen tiedostopolut puuttuvat joten nekin
määritetään.", vbInformation, "Tiedostopolkujen vaihto")

End If

Tiedostopolut()

End Sub

Sub VaihdatiedostopolutValvomo()

If bOnOff = True Then
    MsgBox("Ohjelma pysähtyy koska tiedostopolut vaihdetaan.", vbInforma-
tion, "Tiedostopolkujen vaihto")
End If

bOnOff = False
Label14.Text = "Off"
Label2.BackColor = Color.Black
Label10.Text = "Seis"
ReDim TiedostopolutReadValvomo(0)
TiedostopolutReadValvomo(0) = ""
TextBox3.Clear()
sTiedostopolkuWriteValvomo = ""
TextBox4.Clear()

If sTiedostopolkuReadRobex = "" Or sTiedostopolkuWriteRobex = "" Then

    MsgBox("Myös Rob-Ex:n tiedostojen tiedostopolut puuttuvat joten nekin
määritetään.", vbInformation, "Tiedostopolkujen vaihto")

End If

Tiedostopolut()

End Sub

Sub Aikavali()

'Ajastimen aikavälin valinta - ERROR HANDLER -> NaN
```

'Sekunnit:

```
If CheckBox5.Checked = True Then  
    iAikavali = 100
```

```
Else
```

```
    If RadioButton1.Checked = True Then
```

```
        NumericUpDown1.Minimum = 5  
        NumericUpDown1.Maximum = 60  
        NumericUpDown1.Increment = 1
```

```
        'Vaihdetaan liian suuri tai pieni arvo rajojen sisälle
```

```
        If NumericUpDown1.Value < NumericUpDown1.Minimum Then
```

```
            NumericUpDown1.Value = 1
```

```
        End If
```

```
        If NumericUpDown1.Value > NumericUpDown1.Maximum Then
```

```
            NumericUpDown1.Value = 60
```

```
        End If
```

```
        iAikavali = (1000 * NumericUpDown1.Value)
```

```
    End If
```

'Minuutit:

```
If RadioButton2.Checked = True Then
```

```
    NumericUpDown1.Minimum = 1  
    NumericUpDown1.Maximum = 60  
    NumericUpDown1.Increment = 1
```

```
    'Vaihdetaan liian suuri tai pieni arvo rajojen sisälle
```

```
    If NumericUpDown1.Value < NumericUpDown1.Minimum Then
```

```
        NumericUpDown1.Value = 1
```

```
    End If
```

```
    If NumericUpDown1.Value > NumericUpDown1.Maximum Then
```

```
        NumericUpDown1.Value = 60
```

```
    End If
```

```
    iAikavali = (1000 * NumericUpDown1.Value * 60)
```

```
End If

'Tunnit:
If RadioButton3.Checked = True Then

    NumericUpDown1.Minimum = 1
    NumericUpDown1.Maximum = 24
    NumericUpDown1.Increment = 1

    'Vaihdetaan liian suuri tai pieni arvo rajojen sisälle
    If NumericUpDown1.Value < NumericUpDown1.Minimum Then

        NumericUpDown1.Value = 1

    End If

    If NumericUpDown1.Value > NumericUpDown1.Maximum Then

        NumericUpDown1.Value = 24

    End If

    iAikavali = (1000 * NumericUpDown1.Value * 120)

End If
End If
End Sub

Private Sub NumericUpDown1_ValueChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles NumericUpDown1.ValueChanged

    'Pysäytetään timeri ja vaihdetaan arvo jos arvoa muutetaan
    Timer1.Stop()

    Aikavali()

    Timer1.Interval = iAikavali

    Timer1.Start()

End Sub

'Tästä alaspäin nappuloita

Private Sub LopetaToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles LopetaToolStripMenuItem.Click

    'Laittaa ohjelman kiinni ylävalikon kautta
```

```
        Close()

    End Sub

    Private Sub KäyttöohjeetToolStripMenuItem_Click(ByVal sender As Sys-
tem.Object, ByVal e As System.EventArgs) Handles KäyttöohjeetToolStripMenu-
Item.Click

        'Ohjeet ohjelman käyttämiseen PDF -tiedostossa - ERROR HANDLER -> OFF

        Try
            Dim sOhjetiedostopolku As String = My.Application.Info.DirectoryPath

            Process.Start(sOhjetiedostopolku + "\ReadWriterOhje.pdf")

        Catch ex As Exception

            MsgBox("Ohjetiedostoa ReadWriterOhje.pdf ei löytynyt ohjelman omasta kansios-
ta!", vbExclamation, "Ohje puuttuu!")

        End Try

    End Sub

    Private Sub PäivitäToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PäivitäToolStripMenuItem.Click

        'Tietojen päivitys ylävalikon kautta

        Paivitatiedot()

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles Button2.Click

        'Tietojen päivitys nappulan kautta

        Paivitatiedot()

    End Sub

    Private Sub OnOffToolStripMenuItem1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles OnOffToolStripMenuItem1.Click

        'Automaatti päälle ylävalikon kautta

        OnOff()

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As Sys-
tem.EventArgs) Handles Button1.Click

        'Automaatti päälle nappulan kautta
```

```
OnOff()
```

```
End Sub
```

```
Private Sub KaikkiToolStripMenuItem_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles KaikkiToolStripMenuItem.Click
```

```
'Kaikkien tiedostopolkujen vaihto ylävalikon kautta
```

```
Vaihdattiedostopolut()
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As Sys-  
tem.EventArgs) Handles Button3.Click
```

```
'Kaikkien tiedostopolkujen vaihto nappulan kautta
```

```
Vaihdattiedostopolut()
```

```
End Sub
```

```
Private Sub RobExToolStripMenuItem_Click(ByVal sender As System.Object, ByVal  
e As System.EventArgs) Handles RobExToolStripMenuItem.Click
```

```
'Rob-Ex tiedostopolkujen vaihto ylävalikon kautta
```

```
VaihdattiedostopolutRobex()
```

```
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As Sys-  
tem.EventArgs) Handles Button4.Click
```

```
'Rob-Ex tiedostopolkujen vaihto nappulan kautta
```

```
VaihdattiedostopolutRobex()
```

```
End Sub
```

```
Private Sub ValvomoToolStripMenuItem_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles ValvomoToolStripMenuItem.Click
```

```
'Valvomon tiedostopolkujen vaihto ylävalikon kautta
```

```
VaihdattiedostopolutValvomo()
```

```
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As Sys-  
tem.EventArgs) Handles Button5.Click
```

```
'Valvomon tiedostopolkujen vaihto ylävalikon kautta
```

```
VaihdattiedostopolutValvomo()
```

```
End Sub
```

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
```

```
'Radiobutton aktivoitessa vaihdetaan halutut arvot
```

```
Timer1.Stop()
```

```
Aikavali()
```

```
Timer1.Interval = iAikavali
```

```
Timer1.Start()
```

```
End Sub
```

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
```

```
'Radiobutton aktivoitessa vaihdetaan halutut arvot
```

```
Timer1.Stop()
```

```
Aikavali()
```

```
Timer1.Interval = iAikavali
```

```
Timer1.Start()
```

```
End Sub
```

```
Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged
```

```
'Radiobutton aktivoitessa vaihdetaan halutut arvot
```

```
Timer1.Stop()
```

```
Aikavali()
```

```
Timer1.Interval = iAikavali
```

```
Timer1.Start()
```

```
End Sub
```

```
'Tiedot ohjelmasta
```

```
Private Sub OhjelmastaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OhjelmastaToolStripMenuItem.Click
```

```
MsgBox(Chr(10) & "XML-CSV ReadWriter for Rob-Ex v1.0" & Chr(10) & Chr(10) &  
Chr(10) & Chr(10) & " Metropolia ammattikorkeakoulu" & Chr(10) & Chr(10) & "  
2013" & Chr(10) & Chr(10) & " by" & Chr(10) & Chr(10) &  
& " Antti Ruohio", vbInformation, "Tiedot")
```

```
End Sub
```



```
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox1.CheckedChanged
    'Kirjoitettavan .csv -tiedoston tiedostopolun nollaus

    If TiedostopolutReadValvomo(0) IsNot "" And sTiedostopolkuWriteValvomo
IsNot "" Then

MsgBox("Valvomon tiedostojen kirjoituksen ja lukemisen tyyppiä vaihdetaan. Luet-
tavien ja kirjoitettavien .csv -tiedostojen tiedostopolkuja täytyy muuttaa.",
vbInformation, "Valvomon tiedonvälityksen vaihto")

    End If

    ReDim TiedostopolutReadValvomo(0)
    sTiedostopolkuWriteValvomo = ""
    TextBox4.Text = ""
    TextBox3.Text = ""
    bOnOff = False
    Label4.Text = "Off"
    Label2.BackColor = Color.Black
    Label10.Text = "Seis"

End Sub

Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox3.CheckedChanged

If TiedostopolutReadValvomo(0) IsNot "" And sTiedostopolkuWriteValvomo IsNot ""
Then

MsgBox("Operaatioiden päivitysmenetelmää vaihdetaan. Luettavien .csv -tiedostojen
tiedostomuoto muuttuu joten uudet tiedostot täytyy määrittää.", vbInformation,
"Operaatioiden päivitysmenetelmän vaihto")

    End If

    ReDim TiedostopolutReadValvomo(0)
    sTiedostopolkuWriteValvomo = ""
    TextBox4.Text = ""
    TextBox3.Text = ""
    bOnOff = False
    Label4.Text = "Off"
    Label2.BackColor = Color.Black
    Label10.Text = "Seis"

    If CheckBox3.Checked = True Then
        CheckBox4.Enabled = True
    Else
        CheckBox4.Enabled = False
        CheckBox4.Checked = False
    End If
End Sub

Private Sub CheckBox5_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox5.CheckedChanged

    'Otetaan timerin säätömahdollisuudet päälle ja pois tilanteen mukaan sekä
säädetään aikaväli oikein.
```

```
If CheckBox5.Checked = False Then

    RadioButton1.Enabled = True
    RadioButton2.Enabled = True
    RadioButton3.Enabled = True
    NumericUpDown1.Enabled = True

    Timer1.Stop()

    Aikavali()

    Timer1.Interval = iAikavali

    Timer1.Start()

Else

    RadioButton1.Enabled = False
    RadioButton2.Enabled = False
    RadioButton3.Enabled = False
    NumericUpDown1.Enabled = False

    Timer1.Stop()

    Aikavali()

    Timer1.Interval = iAikavali

    Timer1.Start()

End If

End Sub

End Class
```

## **TUOTANTOAUTOMAATIO**

### **LABORATORIOHARJOITUS / ROB-EX TUOTANNONSUUNNITTELUJÄRJESTELMÄ**

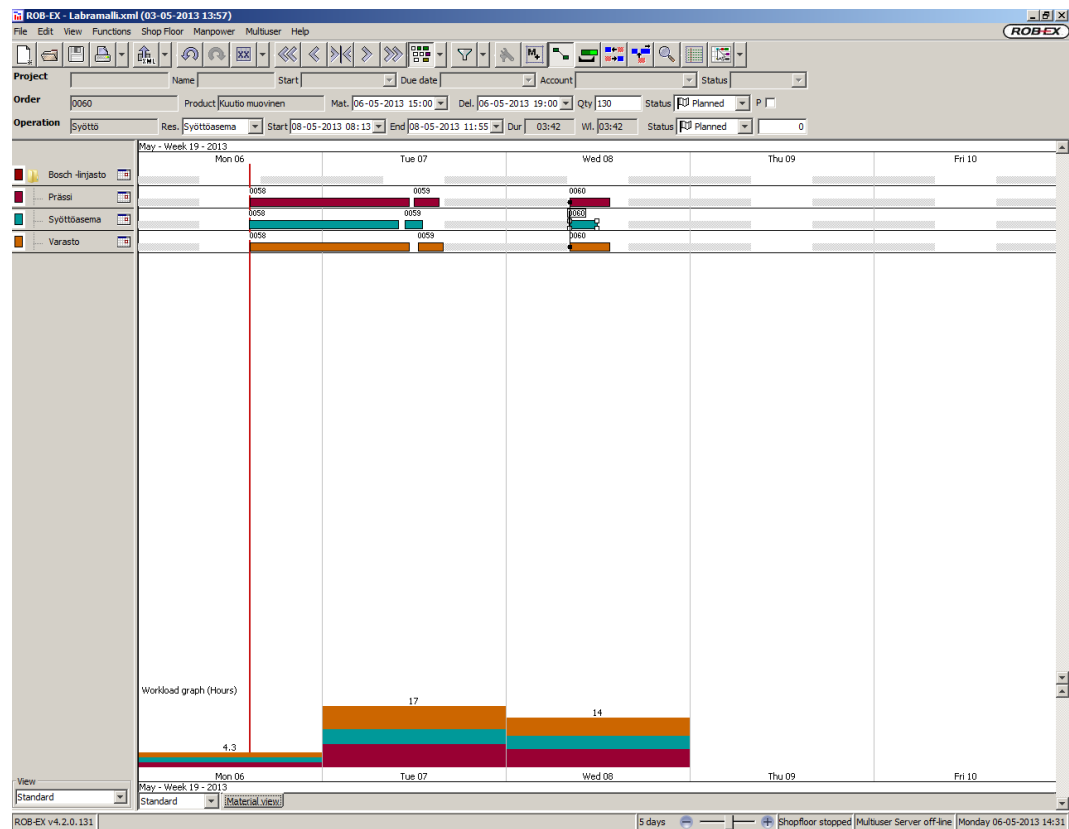
#### **SISÄLLYSLUETTELO**

- 1. YLEISTÄ**
- 2. TYÖOHJE**
- 3. RAPORTTI**

## 1. YLEISTÄ

Tässä osassa työtä luodaan **Rob-Ex** -tuotannosuunnittelujärjestelmään laboratorion kappaletavaralinjaa vastaava malli, simuloidaan ERP -järjestelmän liityntää **Rob-Ex**:ään kevyesti, suunnitellaan tuotantoa **Rob-Ex**:lla ja lopuksi ajetaan tilaus automaatiolaitteistolla.

**Rob-Ex** on **Novotekin** kehittämä erityisesti tuotannon suunnitteluun tarkoitettu **MES** -järjestelmän osana toimiva ohjelmisto, jonka avulla tuotantotilausten aikataulutusta ja tuotannon optimointia onnistuu havainnollisesti **Gantt** -taulukon avulla. Tässä työssä tutustutaan hieman ohjelmiston eri toimintoihin ja integraation periaatteisiin **ERP** -järjestelmän ja tehdastason automaation välillä.



Kuva 8: Rob-Ex käyttöliittymä ja työn tavoite.

Ennen työn aloittamista valvomosovelluksen ja logiikkaprojektin tulee olla toimintakunnossa ja yhteensopivia **Rob-Ex**:in tiedonvälityksen kanssa.

## 2. TYÖOHJE

### 2.1 Rob-Ex konfigurointi

#### 2.1.1 Valmistelut

Aloitetaan työ avaamalla **Rob-Ex Gantt Scheduler** ja luomalla uusi mallipohja laitteisto-  
toa varten.

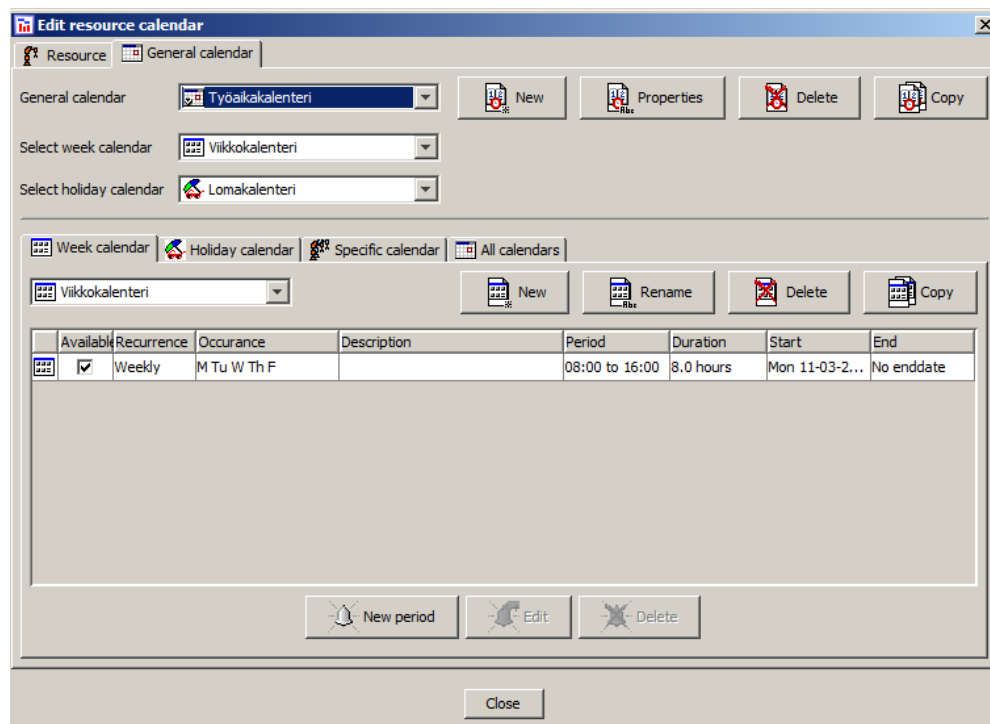
- Tämä tapahtuu valitsemalla ylävalikosta **File → New**. Tietoja ei siirretä vanhasta pohjasta uuteen ja uusi pohja nimetään osuvasti ja tallennetaan vaikkapa työpöydälle.

#### 2.1.2 Kalenteri

Kalenterin määrittämisen ideana on luoda ohjelmalle resurssikohtaiset tai yleiset työ-  
aika- ja lomakalenterit, jotta ohjelma osaa sijoittaa tilaukset suoritettaviksi työaikana,  
eikä silloin sunnuntain vastaisena yönä loma-aikaan. Nyt luodaan vain yksinkertainen  
työaikaa kuvaava kalenteri jotta työ etenee tehokkaasti.

Kalenteri määritetään valitsemalla ylävalikosta **Edit → Calendar**, jolloin avautuvan ik-  
kunan välilehdillä voidaan määrittää yleinen- sekä lomakalenteri.

- Määritetään uusi yleinen kalenteri valitsemalla **General calendar** -välilehdellä **New** ja nimetään asianmukaisesti esim. *"Työaikakalenteri"*.
- Luodaan viikkokalenteri valitsemalla alemmaa ikkunasta **Week calendar** -  
välilehdeltä nappula **New** ja nimetään asianmukaisesti esim. *"Viikkokalenteri"*.
- Alhaalta ikkunan alalaidasta valitaan vielä **New period**, jolloin halutut työajat  
voidaan kalenteriin asettaa. **Description** -kohtaan voidaan kuvata kalenteria  
asianmukaisella termillä esim. *"Työtunnit"*.



Kuva 9: Työaikakalenteri

### 2.1.3 Materiaalit

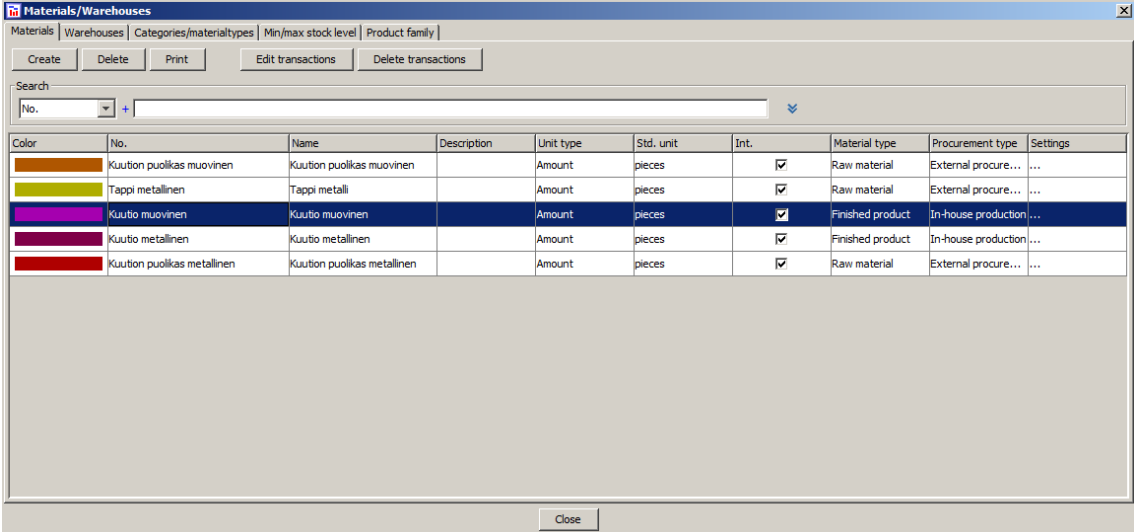
Määrittäessä materiaalit ohjelmistolle pystyy se laskemaan kuluneiden raaka-aineiden ja tuotettujen puolivalmisteiden sekä valmiiden tuotteiden lukumääriä. Tästä on hyötyä, mikäli ohjelma on yhteydessä varastonhallintajärjestelmään. Tällöin tiedot siirtyvä varastolle, missä voidaan suorittaa tarvittavia toimenpiteitä esimerkiksi siten, että tavaraa on varastossa aina optimaalinen määrä.

Linjaston käyttämät materiaalit määritetään valitsemalla ylävalikosta **Edit → Materials**, jolloin avautuvan ikkunan välilehdillä voidaan määrittellä erilaisia materiaaleihin ja tuotteisiin liittyviä ominaisuuksia.

- Määritetään uusi materiaali valitsemalla **Materials** -välilehdellä **Create** ja nimitään asianmukaisesti\*.
- Materiaalin numero ja nimi voivat olla selkeyden vuoksi tässä tapauksessa samoja. Kuitenkaan ne eivät saisi sisältää pilkkuja.

- Materiaaleille voidaan määrittää niiden suure **Unit type** -sarakeeseen sekä yksikkö **Std. Unit** -sarakeeseen. Tässä tapauksessa suure on määrä **Amount** ja yksikkö kappaletta **pieces**, koska kyseessä ovat kiinteät yksittäiset osat. **Int** -kohtaan laitetaan täppä, jotta kappalemäärät esitetään kokonaislukuina.
- Materiaalien materiaalityypeiksi määritetään **Material type** -sarakeeseen joko raaka-aine **Raw material** tai valmis tuote **Finished Product**. Puolivalmisteita ei linjaston välivarastojen uupussa oikeastaan ole.
- **Procurement type** -sarakeeseen voidaan määrittää tulevatko materiaalit esim. alihankkijoilta vai tuotetaanko ne itse.
- Toistetaan edellisiä kunnes kaikki materiaalit määritelty.

*\*Tarkoituksena siis luoda raaka-aineet joista linjan palikat kasataan, ja valmis tuote joka voi olla joko muovia tai metallia.*



The screenshot shows a software window titled "Materials/Warehouses" with a menu bar (Materials, Warehouses, Categories/materialtypes, Min/max stock level, Product family) and buttons for Create, Delete, Print, Edit transactions, and Delete transactions. Below the buttons is a search field with a dropdown menu set to "No." and a search input field. The main area contains a table with the following data:

Color	No.	Name	Description	Unit type	Std. unit	Int.	Material type	Procurement type	Settings
		Kuution puolikas muovinen	Kuution puolikas muovinen	Amount	pieces	<input type="checkbox"/>	Raw material	External procure...	...
		Tappi metallinen	Tappi metalli	Amount	pieces	<input checked="" type="checkbox"/>	Raw material	External procure...	...
		Kuutio muovinen	Kuutio muovinen	Amount	pieces	<input checked="" type="checkbox"/>	Finished product	In-house production	...
		Kuutio metallinen	Kuutio metallinen	Amount	pieces	<input checked="" type="checkbox"/>	Finished product	In-house production	...
		Kuution puolikas metallinen	Kuution puolikas metallinen	Amount	pieces	<input checked="" type="checkbox"/>	Raw material	External procure...	...

Kuva 10: Materiaali

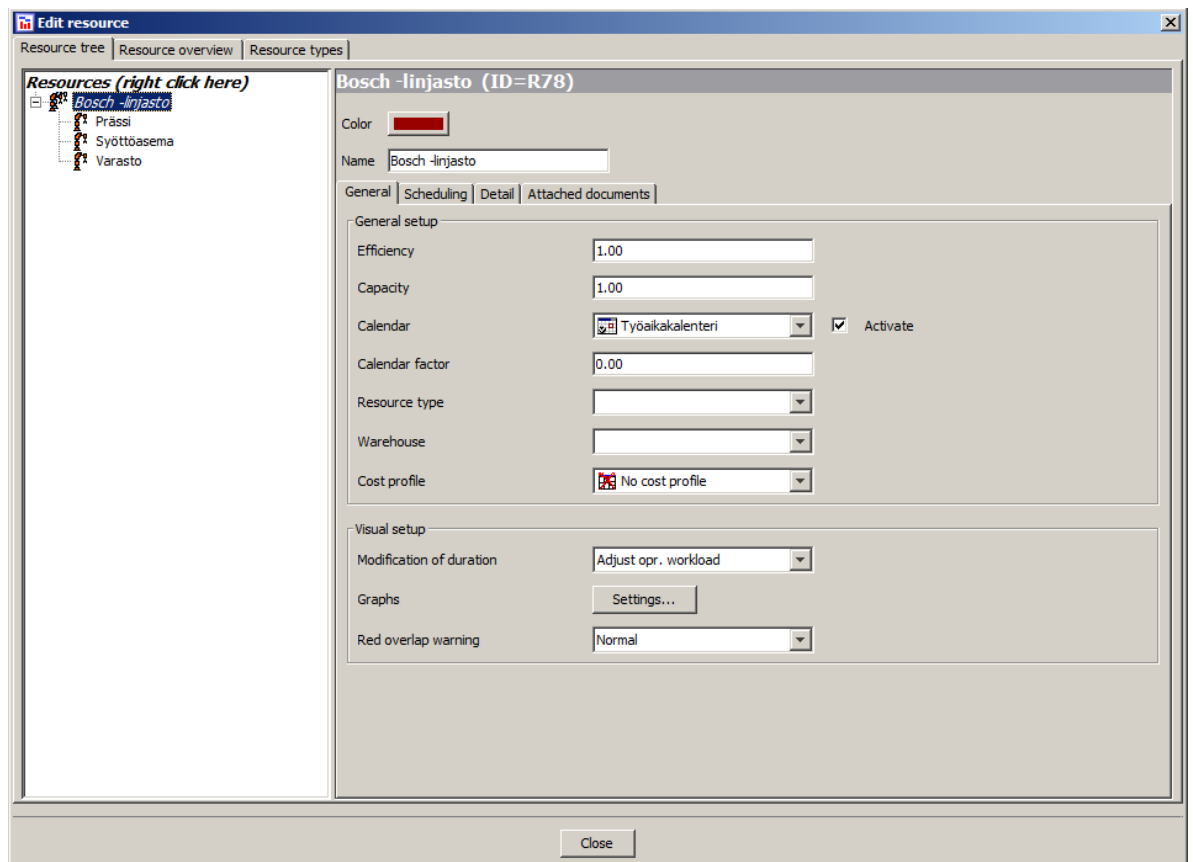
## 2.1.4 Resurssit

Resurssit kuvaavat käytössä olevaa työvoimaa, eli yleensä yrityksen työntekijöitä tai koneita.

Tuotantoyksikön resurssit määritetään valitsemalla ylävalikosta **Edit → Resource**, jolloin avautuvassa ikkunassa voidaan määrittellä käytettävät resurssit ja niiden työtehdokkuus ym. lisätietoja.

- Ikkunan vasemmalla olevaan valkoiseen kenttään voidaan lisätä resursseja ja ryhmitellä niitä halutulla tavalla. Lisätään ensin resurssiryhmä klikkaamalla hiiren oikealla ja valitsemalla **Add Resource Group**. Resurssiryhmä nimetään asianmukaisesti esim. ”Linjasto” ja se asetetaan käyttämään luotua työaikakalenteria **Calendar** -vetovalikosta valitsemalla ja **Activate** -täppäämällä.
- Lisätään luotuun resurssiryhmään resursseja\* samalla menetelmällä valitsemalla **Add Resource** ja osuvasti nimeämällä, sekä työaikakalenteri määrittämällä.
- Toistetaan edellistä kunnes kaikki resurssit määritelty.

*\*Tässä työssä linjasto jaetaan yksinkertaistetusti kolmeen osaan, joista ensimmäinen syöttää palikoita, toinen prässää ne ja kolmas siirtää valmiit tuotteet varastoon.*



Kuva 11: Resurssit



### 2.1.5 Operaatiot

Operaatiot kuvaavat tuotteiden tekemisessä suoritettavia työvaiheita.

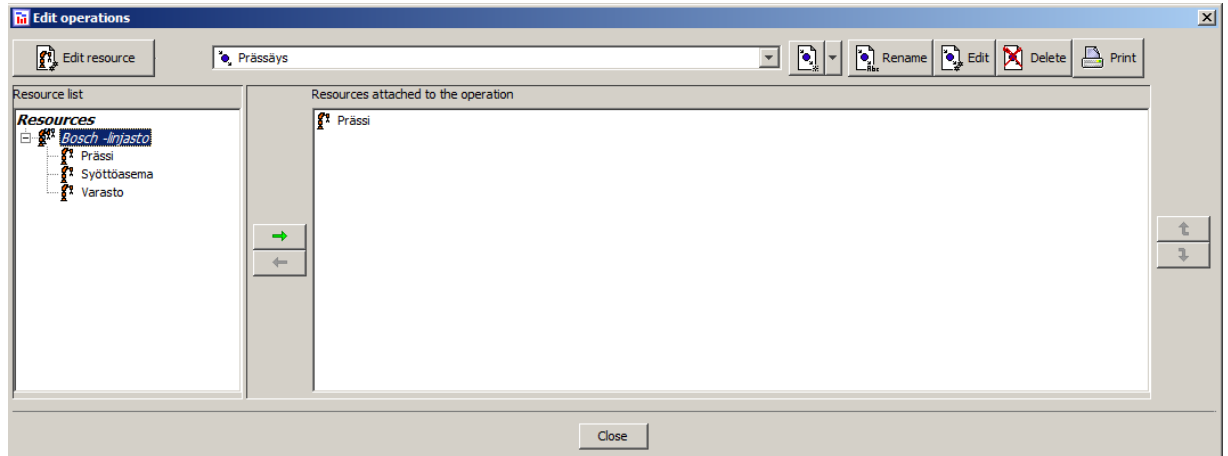
Tuotantoyksikön operaatiot määritellään valitsemalla ylävalikosta **Edit → Operations**, jolloin avautuvissa ikkunoissa voidaan määritellä suoritettavat operaatiot ja niiden kuluttamat tai tuottamat materiaalit.



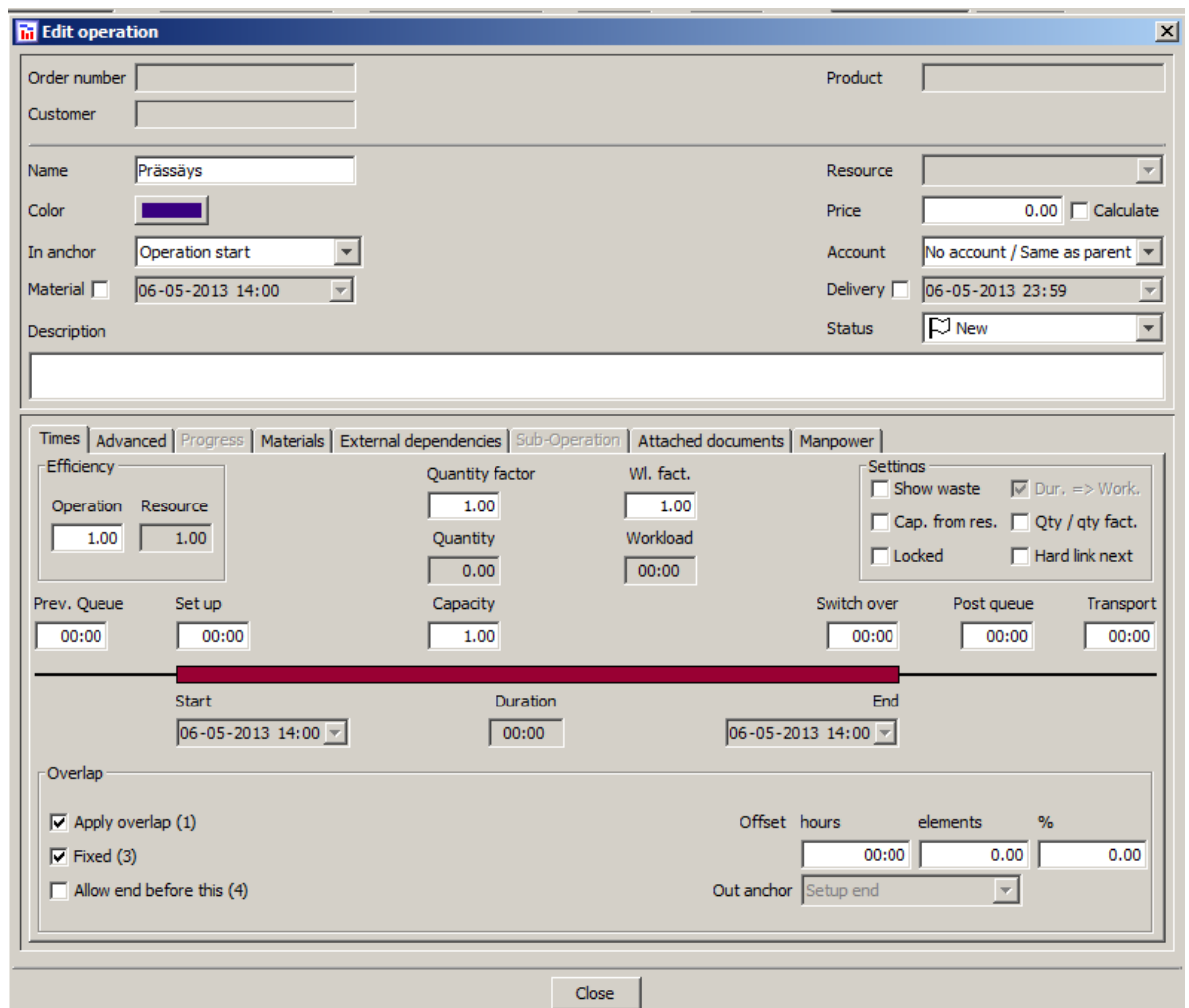
- Määritetään uusi operaatio\* valitsemalla **Add New Operation** ja nimitetään operaatio asianmukaisesti.
- Siirretään vasemmasta kentästä operaation suorittava resurssi oikeaan kenttään hiirellä valitsemalla ja vihreää nuolta painamalla.
- Operaatiot määritetään vielä siten, että ne voivat tapahtua suunnitelmassa samanaikaisesti.\*\* Valitaan **Edit** ja avautuvassa **Edit operation** -ikkunassa asetetaan täpät kohtiin **Apply Overlap** ja **Fixed**.
- Samalla määritellään **Edit operation** -ikkunan **Materials** -välilehdellä mitä materiaaleja ko. operaatio kuluttaa tai tuottaa.
- Toistetaan edellisiä kunnes kaikki operaatiot määritetty.

*\*Tässä työssä operaatiot jaetaan yksinkertaistetusti kolmeen osaan, joista ensimmäinen on palikoiden syöttö, toinen palikoiden prässäys ja kolmas valmiiden palikoiden siirtäminen varastoon.*

*\*\*Linjaston eri laitteiden työvaiheet tapahtuvat käytännössä melkein samanaikaisesti joten eri operaatioiden voidaan ajatella yksinkertaisesti valmistuvan lähes samanaikaisesti.*



Kuva 12: Operaatioiden määrittäminen



Kuva 13: Operaatioiden muokkaaminen

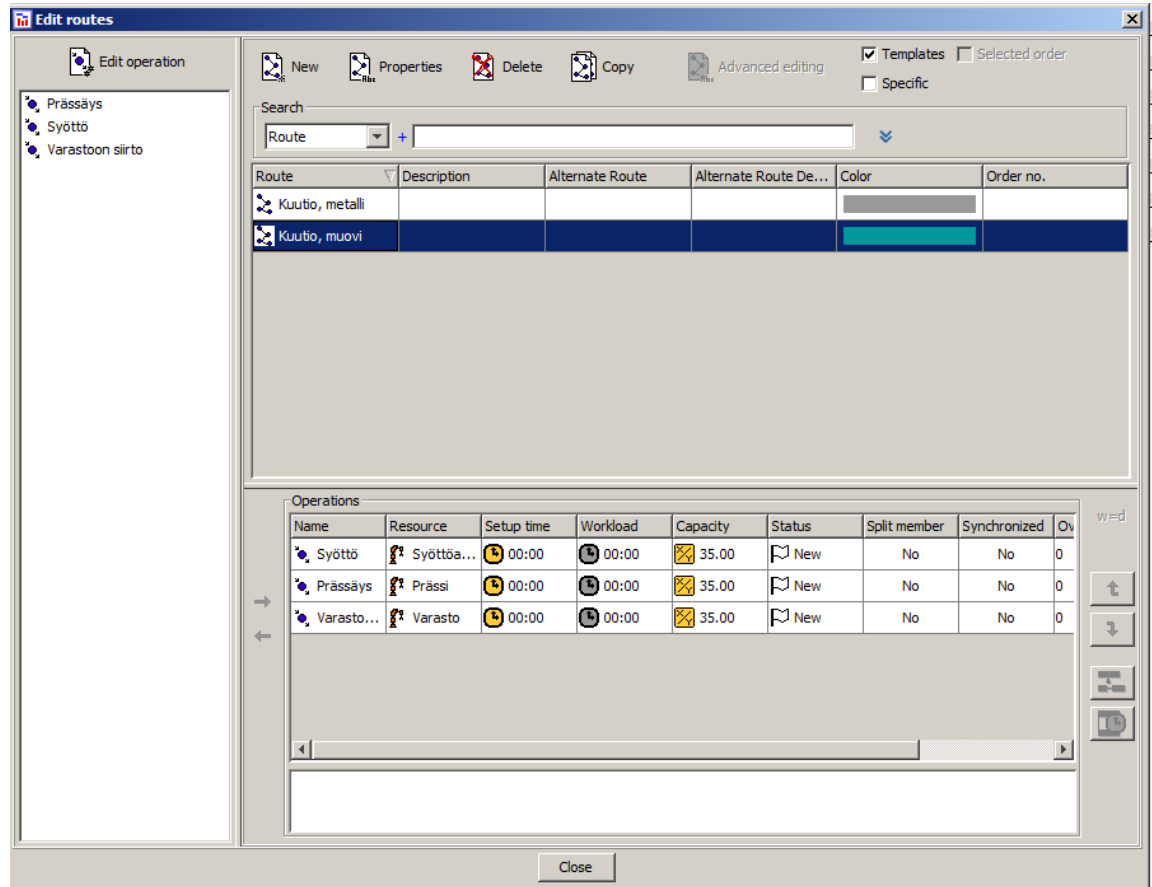
### 2.1.6 Reitit

Reitit kuvaavat mitä kaikkia operaatioita eli toimenpiteitä tulee suorittaa tuotetta valmistuksessa ja ne luodaankin täten tuotekohtaisesti.

Reitit voidaan määrittää valitsemalla ylävalikosta **Edit → Routes**, jolloin avautuvassa ikkunassa voidaan määrittää reitit ja niihin liittyvät operaatiot.

- Määritetään uusi reitti valitsemalla **New** ja nimetään asianmukaisesti esimerkiksi valmistettavan tuotteen mukaan.
- Siirretään reittiin kuuluvat operaatiot vasemmasta kentästä oikealla alhaalla olevaan kenttään reitin ollessa valittuna hiirellä operaatio valitsemalla ja vihreää nuolta painamalla.
- Määritetään operaatioiden kapasiteetti\* **Capacity** -sarakkeeseen. Tässä tapauksessa montako kappaletta valmistuu tunnissa.
- Toistetaan kunnes reitit määritelty.

*\*Koska välivarastoja ei linjastolla ole ja operaatiot suoritetaan näennäisesti samaan aikaan, voidaan kapasiteetti määrittää kaikkiin operaatioihin tuotekohtaisesti samansuuruisiksi.*



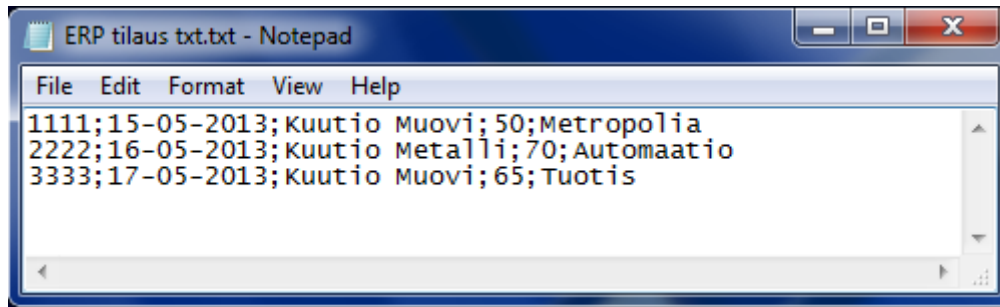
Kuva 14: Reittien määrittäminen

## 2.2 ERP -järjestelmän tilausten lukeminen Rob-Ex:ään

### 2.2.1 Tilaustiedoston luominen

ERP -järjestelmää simuloidaan kevyesti luomalla tekstimuotoinen tiedosto josta ”ERP”:n määrittämiä tilauksia tuodaan **Rob-Ex**:ään.

- Tehdään työpöydälle tekstimuotoinen tiedosto johon määritellään kolme tilausta esimerkkikuvan mallin mukaisesti. Tilausten tuotteet nimetään Rob-Ex:ään luodun mallin tuotenimien mukaan. Myös päivämäärä kannattaa merkitä päivän tai pari myöhemmäksi kuin tämänhetkinen päivämäärä\*.



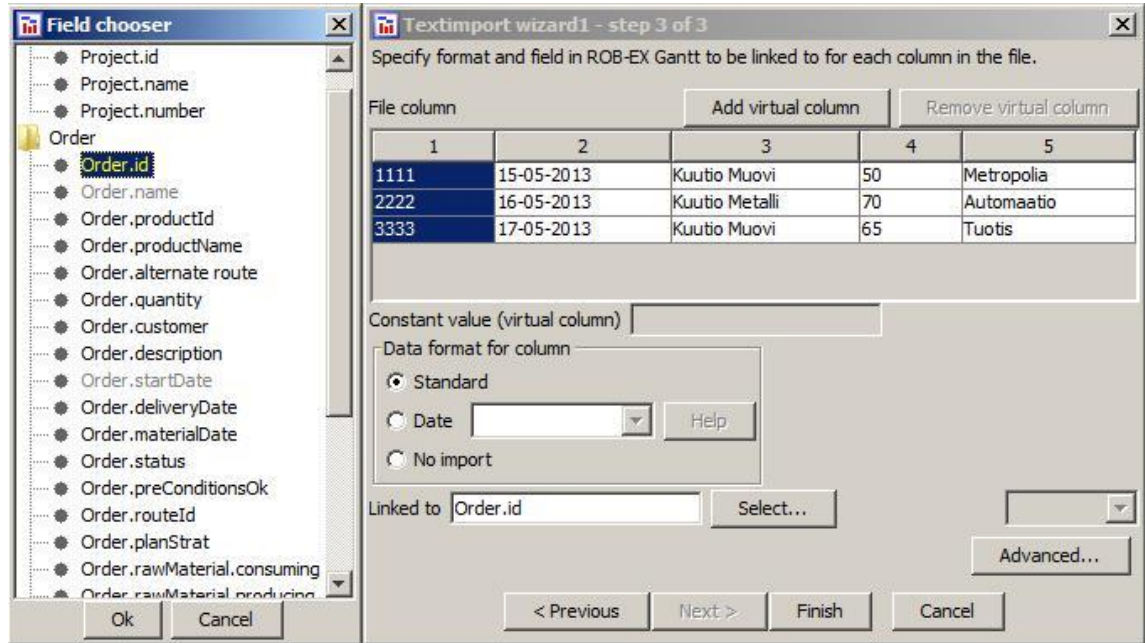
Kuva 15: Esimerkki tilaustiedostosta

\*Ohje tehty menneisyydessä, joten esimerkkikuvakin on muinainen suhteessa nykyhetkeen.

## 2.2.2 Tilaustiedoston lukeminen Rob-Ex:ään

Tekstiedostosta voidaan lukea sinne kirjoitetut tilaukset määrittämällä eli määrittämällä Rob-Ex:lle tiedoston rakenne ja tuomalla tiedot sitten Rob-Ex:n tilauslistalle.

- Tiedoston määrittäminen suoritetaan valitsemalla **File → Import → Textfile (CSV1)**, valitsemalla avautuvasta ikkunasta **New** ja nimeämällä määrittäminen osuvasti. Tämän jälkeen valitaan alla olevaan sarakkeeseen työpöydälle luotu tilaustiedosto ja painetaan **Next**.
- Seuraavassa kohdassa määritetään tiedot erottelava merkki, joka esimerkkitiedoston tapauksessa oli puolipiste **Semicolon**. Määrittämällä erottelumerkki **Rob-Ex** osaa erotella tiedoston sisältämät tiedot sarakkeisiin.
- Seuraavaksi määritellään mitkä sarakkeet linkitetään mihinkin muuttujiin **Rob-Ex**:n tilaustietokannassa.
- Valitaan tilausnumerot sisältävä sarake, kohta **Standard** ja linkitetään **Order.id**-muuttujaan.
- Valitaan päivämäärän sisältävä sarake, kohta **Date (dd-MM-yyyy)** ja linkitetään **Order. Deliverydate**
- Valitaan tuotenimen sisältävä sarake, kohta **Standard** ja linkitetään **Order.Productname**-muuttujaan.
- Valitaan valmistettavan kappalemäärän sisältävä sarake, kohta **Standard** ja linkitetään **Order.Quantity** -muuttujaan.
- Valitaan asiakkaan sisältävä sarake, kohta **Standard** ja linkitetään **Order.Customer** -muuttujaan.
- Kuitataan määrittäminen valmiiksi **Finish** ja valitaan luetaan tilaustiedosto käyttäen edellä tehtyä määrittäystä.



Kuva 16: Tilaustiedoston mäppäys

## 2.3 Tilausten suunnittelu tuotantoa varten

### 2.3.1 Tilausten suunnittelu

**Rob-Ex**:ssa tilauksia voidaan suunnitella joustavasti ja havainnollisesti ohjelmalle omista **Gantt** -taulukkoa ja erilaisia valmiita suunnittelutoimintoja hyväksikäyttäen.

- Suunnitellaan tilaustiedostosta löytyvät tilaukset valitsemalla ylävalikosta **Edit** → **Order list**.
- Tarkistetaan tilaukset valitsemalla **Edit** ja asetetaan tiedot oikein, mikäli on tarpeen. Tilaukselle voidaan määrittää tässä ikkunassa kaikki tarvittavat tiedot ja valmistettavaa tuotetta vastaava reitti.
- Tehdään sama kaikille tilauksille kunnes ovat **Order list** -ikkunan **Status** -sarakeessa tilassa **New**.
- Suunnitellaan tilaukset siten, että ne tuotetaan oikeassa järjestyksessä vaihtamalla hiirellä klikkailemalla **Status** -sarakeessa valkoisen **New** -lipun tilaan **Planned** -tilaan.
- Ohjelma kysyy miten valittu tilaus suunnitellaan. Tässä valitaan tilaukset suunniteltavaksi toimintoja hyväksikäyttäen siten, että ne tulevat tuotetuiksi ajallaan ja järkevästi.

**Edit order**

Order number: 1111

Series: No series

Customer: Metropolia

Product: Kuutio Muovi

Quantity: 50

Account: No account / Same as parent

Project: No project

Materials ready: 17-05-2013 12:11

Delivery date: 18-05-2013 15:00

Route: Kuutio, muovi

Operations

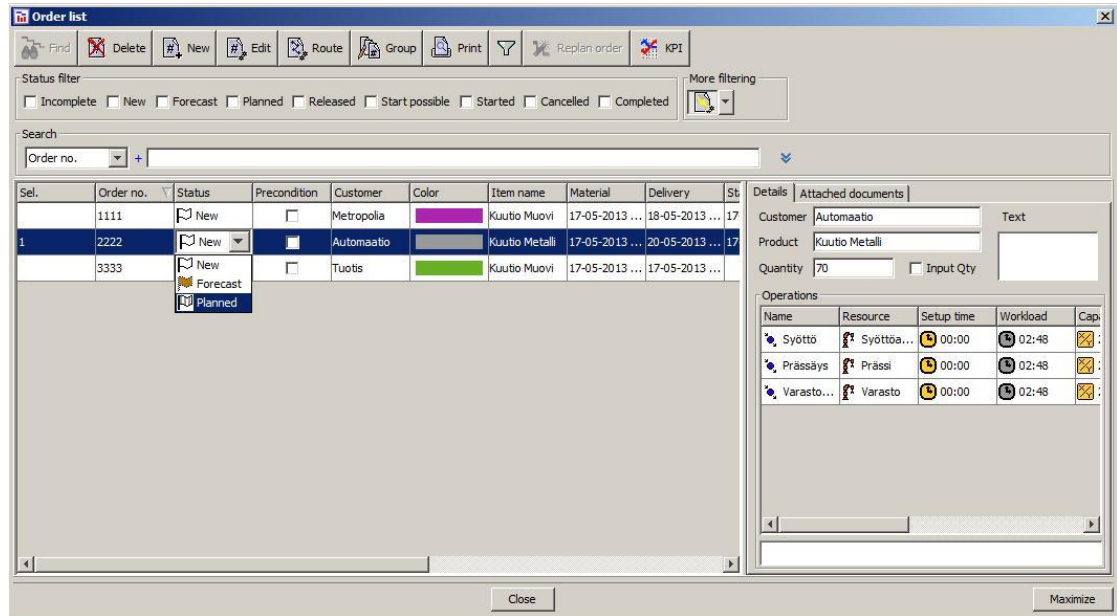
Name	Resource	Setup time	Workload	Capacity	Status	Split m...
Syöttö	Syöttöasema	00:00	00:00	35...	New	No
Prässäys	Prässi	00:00	00:00	35...	New	No
Varastoon siirto	Varasto	00:00	00:00	35...	New	No

Create planned

Scheduling strategy

Create and stay Ok Cancel

Kuva 17: Tilaustietojen editointi



Kuva 18: Tilauslista

### 2.3.2 Toiminnot

Rob-Ex sisältää useita eri suunnittelutoimintoja, tässä osa hyödyllisistä toiminnoista.

- Tilauksia voidaan suunnitella uudelleen hiiren oikealla **Gantt** -taulukossa jonkin operaation päällä **Re-plan Order** valitsemalla.
- Suoritettavia operaatioita voidaan järjestellä manuaalisesti **Gantt** -taulukossa valitsemalla ja hiirellä raahaamalla.
- Operaatioita voidaan jakaa hiiren oikealla **Gantt** -taulukossa jonkin operaation päällä **Split and synchronize** → **Split operation**.
- Operaation suorittavaa resurssia voidaan vaihtaa hiiren oikealla **Gantt** -taulukossa jonkin operaation päällä **Change resource** valitsemalla.
- Hitaammin tuotantoa suorittavien resurssien aiheuttamia pullonkaloja voidaan selvittää hiiren oikealla **Gantt** -taulukossa selvitettävät operaatiot valittuna **Bottleneck Sort** valitsemalla.

### 2.3.3 Uusi tilaus

Saitte juuri tuotteiden valmistusta koskevan suunnitelman työpäivän päätteeksi valmiiksi mutta tärkein asiakkaanne **Rekkula Oy** ilmoitti että tarvitsee **huomiseksi klo 18.00** mennessä **15 -kappaletta muovisia kuutioita** tärkeää toimitusta varten. Teidän tulee



nyt jäädä **ylitöihin** ja muuttaa tuotantosuunnitelmaa siten, että asiakkaanne vaatima tilaus saadaan toteutettua aikataulussa.

- Mikäli käytetään vain linjaston prässä, kannattaa luoda uusi reitti jossa valmistettavat kappaleet vain prässätään.
- Luodaan uusi tilaus **Order List** -tilauslistalle valitsemalla **New** ja sen jälkeen täyttämällä avautuvan ikkunan kenttiin kaikki tarvittavat tilaustiedot ja valitsemalla oikea reitti.
- Suunnitellaan tilaus muuttamalla sen tilaksi **Planned**.
- Suunnitellaan valmistussuunnitelma uusiksi siten että tilaus tulee tehdyksi aikataulussa mahdollisimman vähän muita tilauksia häiriten.

## 2.4 Tilauksen tuottaminen

### 2.4.1 Tilauksen vapauttaminen

Uusi päivä koittaa! Vaikka eilen meni myöhään eikä korvaus ollut kummoinen, olette silti heti aamusta motivoituneita, koska pääsette tuottamaan suunnittelemaanne **Rekkula Oy:n** tilausta.

- Tilaus vapautetaan tuotantoon asettamalla **Order List** -tilauslistalla **Precondition** -sarakkeeseen täppä ja sen jälkeen vaihtamalla tilauksen tila lipun kohdalla siniseksi tilaan **Start Possible**. Sama onnistuu myös **Gantt** -taulukosta tilauksen valitsemalla ja oikealla ylhäällä asettamalla **P** -kohtaan täppä ja vaihtamalla tilan viereisistä vetovalikoista.
- Tiedonsiirtoa varten on luotu erillinen **ReadWriter for Rob-Ex** - tiedonvälitystyökalu, joka välittää tilauksen olennaiset tiedot **CSV** -tiedostona valvomolle kohdekansioon ja lukee valvomon kirjoittamat tiedot samalla menetelmällä. Lisäksi **Rob-Ex**:ään on valmiiksi määritetty makrotoiminnot, joiden avulla tietoa kirjoitetaan ulos ja luetaan sisään automaation suunnalta.
- Valvomon kirjoittamien tiedostojen formaatti on määritelty mm. tiedonsiirtotyökalun ohjeessa.

### **2.4.2 Tilauksen ajaminen**

Tilaustiedot on välitetty valvomon luettavaksi ja operaattori ottaa tilauksen vastaan in-nokkaana. Tilaus asetetaan ajettavaksi välittömästi.

- Tilaus ajetaan automaatiolaitteistolla ja valvomon kirjoittamien tietojen tulisi päivittyä **Rob-Ex**:ään.
- Toimintaa voidaan testata uudelleen suunnittelemalla ajettava tilaus uudelleen.

## **2.5 Lopputoimenpiteet**

### **2.5.1 Turhien tiedostojen poistaminen**

Poistetaan lopuksi turhat tiedostot ettei jää ylimääräisiä sekoittamaan toimintaa.

- Poistetaan luotu mäppäystiedosto **Rob-Ex** kansioista, tai etsitään vaikka haku-toiminnolla että löytyy.
- Poistetaan luotu mallipohja ja avataan **Rob-Ex**:ään oletustiedosto **Labramalli**.

### **2.5.2 Lopetus**

- Suljetaan **Rob-Ex Gantt Scheduler**.

## **3. RAPORTTI**

Yksittäisistä töistä ei tarvitse laatia työselostusta. Laadittu kokonaissovellus esitellään viimeisellä harjoituskerralla muille ryhmille ja laadittu sovellus dokumentoidaan niin, että sen toimintaperiaate ja arkkitehtuuri käy selville.