Expertise
and insight
for the future

Antti Karjalainen

SCADA Application Engineering in Metro and Railway Projects

Metropolia University of Applied Sciences
Bachelor of Engineering
Electrical and automation engineering
Bachelor's Thesis
10 March 2020

Author Title Number of Pages Date	Antti Karjalainen SCADA Application Engineering in Metro and Railway Projects 61 pages 10 March 2020
Degree	Bachelor of Engineering
Degree Programme	Electrical and automation engineering
Professional Major	Automation technology
Instructors	Johan Laine, Project Manager Reijo Leinonen, Senior Lecturer

ABB MicroSCADA Pro control system is designed to be used in substation and process industry applications. However, nowadays the system is also used in metro and railway applications. The purpose of this thesis was to collect and document the special functions used in MicroSCADA Pro metro and railway applications. Finally, the intention was to select one of these special functions and to develop an engineering improvement to it. The thesis has been made for ABB Power Grids Finland Oy.

All special functions created for metro and railway applications and their features were documented by sub-regions. No areas for development were found from the special functions. Instead, an alternative method to create process displays was created alongside the List Based Engineering (LBE) tool. LBE is a Microsoft Excel based tool that uses macros to create databases and process displays into MicroSCADA Pro. Devices and symbols of metro applications were used in the development phase.

The development of the new process display creation method began by modifying the configuration of the Object Browser menu in the MicroSCADA Pro Display Builder to include all devices and symbols used in metro applications. After that, the Tool Launcher configurations that open the control dialog were added to the metro application symbols.

The result of the thesis study is a comprehensive documentation of the special functions used in MicroSCADA Pro metro and railway applications. The developed process display creation method is an efficient solution when the LBE tool cannot be used and it is also applicable to all MicroSCADA Pro applications that use special symbols.

Keywords	MicroSCADA Pro, control room, metro, process display

Tekijä Otsikko Sivumäärä Aika	Antti Karjalainen SCADA-sovellussuunnittelu metro- ja rautatieprojekteissa 61 sivua 10.3.2020
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	automaatiotekniikka
Ohjaajat	projektipäällikkö Johan Laine lehtori Reijo Leinonen

ABB MicroSCADA Pro -valvomojärjestelmä on suunniteltu käytettäväksi sähköasema- ja prosessiteollisuussovelluksissa. Järjestelmää käytetään kuitenkin nykyään myös useissa metro- ja raideliikennesovelluksissa. Opinnäytetyön tavoitteena oli kerätä ja dokumentoida MicroSCADA Pro:ssa käytettäviä erikoistoimintoja, jotka ovat luotu metro- ja raideliikennesovelluksia varten. Dokumentoinnin jälkeen tarkoitus oli valita yksi näistä erikoistoiminnoista ja kehittää sitä paremmaksi. Opinnäytetyö on tehty ABB Power Grids Finland Oy:lle.

Kaikki metro- ja raideliikennesovelluksia varten luodut erikoistoiminnot ja niiden ominaisuudet dokumentoitiin niiden käyttökohteiden mukaan osa-alueittain. Erikoistoiminnot todettiin toimiviksi, joten niiden sijaan päädyttiin kehittämään vaihtoehtoinen prosessikuvien luontitapa List Based Engineering (LBE) -työkalun rinnalle. LBE on Microsoft Excel -pohjainen työkalu, jolla on mahdollista luoda tietokantoja ja prosessikuvia MicroSCADA Pro:hon makrojen avulla. Kehitystyössä käytettiin metrosovelluksissa käytettäviä laitteita ja symboleita.

Uuden prosessikuvien luontitavan kehittäminen aloitettiin muokkaamalla MicroSCADA Pro:n kuvanrakennustyökalussa olevan Object Browser -valikon konfigurointia siten, että se sisältää kaikki metrosovelluksissa käytettävät laitteet ja niiden symbolit. Tämän jälkeen metrosovellusten symboleihin lisättiin ohjausdialogin avaavat Tool Launcher -konfiguraatiot.

Opinnäytetyön tuloksena on kattava kuvaus metro- ja raideliikennesovelluksissa käytettävistä erikoistoiminnoista. Kehitetty prosessikuvien luontitapa nopeuttaa työskentelyä huomattavasti silloin, kun LBE-työkalua ei voida käyttää. Sitä on myös mahdollista käyttää kaikissa MicroSCADA Pro -sovelluksissa, joissa käytetään itse luotuja symboleja.

Avainsanat	MicroSCADA Pro, valvomo, metro, prosessikuva

Contents

List of Abbreviations

CD	Configuration data attribute of the process group in MicroSCADA.
DALI	Digital Addressable Lighting Interface. A control system for lighting which offers a centralized platform for light control.
DB	Double binary input/double indication. A process object type in MicroSCADA, which contains values 0, 1, 2 and 3.
DMS	Distribution Management System
I/O	Input/output
LBE	List Based Engineering tool. A Microsoft Excel -based tool, which is used for creating process objects and process displays to MicroSCADA.
LN	The logical name of the process group that the object belongs to. The individual process objects in the group are identified by indices.
OS	Object Status. An attribute that indicates the reliability of the registered data in MicroSCADA.
OV	The value of the process object as registered in the process database.
PIR	Passive infrared sensor. Electronic sensor that measures infrared light from the environment.
PLC	Programmable Logic Controller

RTU	Remote Terminal Unit
RX	Reserved Text. A process object attribute where text can be saved.
SCADA	Supervisory Control and Data Acquisition
SCIL	Supervisory Control Implementation Language. A programming language created by ABB used only in MicroSCADA.
SCS	Strömberg Control System. A control system created by Strömberg Oy, later renamed MicroSCADA.
SRB	Smoke removal block. Part of the smoke removal system in MicroSCADA metro applications.

1 Introduction

1.1 About this Thesis

The subject of this thesis study was given by ABB Power Grids Finland Oy. ABB Power Grids Finland started as an independent company on November 1, 2019. Moving Power Grids into an independent company was one part of divesting the Power Grids business from ABB to Hitachi. The sale is expected to be completed in the first half of 2020. [1.]

Target of this thesis is to collect and document all special functions and engineering phases needed in MicroSCADA railway and metro applications. After documenting the special functions, one of them is selected and engineering improvement to it is to be planned, developed and documented in a detailed level. Also, a test procedure of the selected function is to be developed and piloted in one of the existing railway or metro projects.

1.2 SCADA Applications in General

SCADA (Supervisory Control and Data Acquisition) system is a process control system essential to a large scale of industries. With a SCADA system it is possible to supervise, control and gather data from the process. SCADA applications have a great scalability and flexibility which makes them a good choice for many industries and different locations. [2.]

SCADA is a multi-level system. The components and levels of a standard SCADA system are shown in Figure 1. The structure starts with field devices. The field devices

are connected to Remote Terminal Units (RTUs) or Programmable Logic Controllers (PLCs) depending on the used hardware. The communication data with field devices is routed from RTUs and PLCs to a supervisory computer, which includes the SCADA software. The communication route can be wired or wireless. The software in the supervisory computer then displays the data coming from the process to the operators in the control room. [3.] The data coming from the process allows the operators to monitor system events in real-time and to react quickly to the changes in the process.

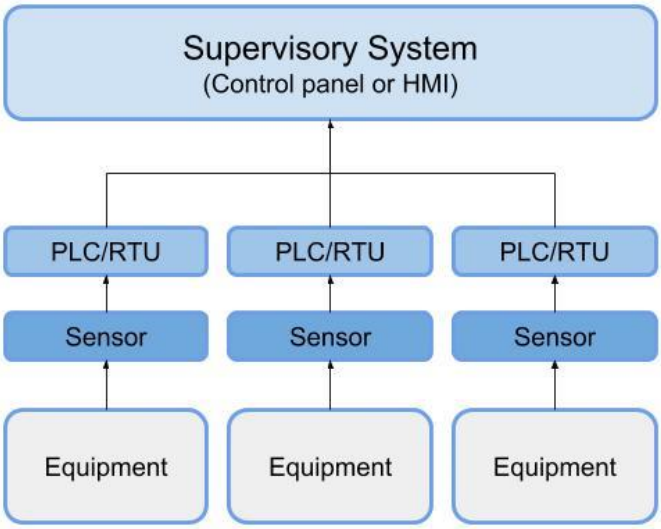

Figure 1. Components of a SCADA system [4].

2 Features of MicroSCADA

2.1 MicroSCADA in General

ABB MicroSCADA is a programmable and distributed SCADA system. It is microcomputer-based, which means it runs on almost every available PC. All MicroSCADA applications and most of its configuration programs are built using SCIL (Supervisory and Control Implementation Language), a programming language used only in MicroSCADA. MicroSCADA is mainly designed for substation automation and process industry applications but it can be used in many other non-electrical applications including heating, water, oil and gas distribution processes. [5.] Nowadays MicroSCADA is also being used in infrastructure applications such as metro and railway systems.

Development of MicroSCADA started in 1982 in Strömberg Oy, a Finnish electrotechnical company. Strömberg wanted to be able to make total automation systems themselves, but they did not have a proper control and reporting system. A new control system was created, and it was named Strömberg Control System (SCS). SCS was later renamed MicroSCADA when Strömberg joined ABB. [6,2.]

MicroSCADA supports all main communication protocols, such as IEC 60870-5-101/104, IEC 61850, ANSI, SPA and Modbus. Communication protocols are used in process communication between system server and process devices. Each protocol has its own characteristics, which means that all interfaces and physical equipment must support the used protocol. [7,20.]

The MicroSCADA Pro product family is focused on power system automation. Currently, it includes the following products:

- MicroSCADA Pro Control System SYS600
- MicroSCADA Pro Compact System SYS600C
- MicroSCADA Pro Distribution Management System DMS600

SYS600 is a programmable and scalable substation automation system. Its most common uses are in electric power processes, but it can also be used for the supervision and control of many other industries as well [8,11]. SYS600C is a pre-configured and compact automation solution. It is a robust industry grade computer designed for all harsh environments [9]. DMS600 is a Distribution Management System which is integrable with SYS600. It offers an advanced DMS database which can be used for improved network monitoring and utilizing the geographical map [10].

The content of this thesis is based only on the SYS600 control system. Therefore, the term MicroSCADA only refers to the SYS600 later in this thesis.

MicroSCADA has an in-built set of object symbols related to power processes. Using this object library greatly speeds up the creation of the database and process displays, because it includes the most used objects in substation and process industry applications.

2.2 MicroSCADA Database

MicroSCADA application requires connection to the process equipment to fulfill its requirements of monitoring and controlling the process devices in certain application area. The process equipment includes system and application objects. System objects

are used for configuring the system and application objects are used for the signals of the process devices. [11,25-26.]

System objects are programmable units that specify the SYS600 communication type and system configuration. System objects are divided into two main types: base system objects and communication system objects. Base system objects are used for the configuration of the base system, whereas communication system objects are used for defining the communication and configuration properties of process communication. [12,13.]

Application objects are programmable units used for different kinds of tasks, for example real time supervision, data registration and calculations. There are eleven types of application objects in MicroSCADA [8,12.]:

- Process objects
- Event handling objects
- Scales
- Data objects
- Command procedures
- Time Channels
- Event Channels
- Logging profiles
- Event objects
- Free Type objects
- Variable objects

Every application object has attributes which store data and information about the object's value, functions, properties etc. All application objects and their attributes are

accessible from the Object Navigator. In the Object Navigator, it is possible to do many object handling functions, such as adding, copying, filtering, exporting and importing application objects [8,249].

2.3 Creation of Control Dialogs

Control dialogs are used to control and monitor the state of a certain device. They contain a lot of different items, for example buttons, texts, menus and images. In MicroSCADA, control dialogs are called Visual SCIL Objects (VSOs). VSOs are created and edited in the Dialog Editor. The Dialog Editor allows the user to design the look of the dialog and its items and to program the dialogs [13,17]. The control dialogs are programmed using methods. Methods are SCIL programs, which can be executed cyclically, by user operation or on a certain event [13,25]. Methods are used to define the operation and behavior of control dialogs.

Control dialogs can be opened by adding a Tool Launcher event to a graphical object in the Display Editor, which is used for the creation of process displays. Tool Launcher is activated when a graphical object with a Tool Launcher event is clicked with mouse in a process display. In addition to VSO-files, Tool Launcher can also be set to open ActiveX tools and other displays as well as to run SCIL codes and command lines [14,47]. Figure 2 shows Tool Launcher settings for a door control dialog. Custom arguments are used in many control dialogs. They are usually meant to specify and describe the controlled object. For example, custom arguments for the door control dialog in Figure 2 include the Logical Name (LN) of the door and a text label which is shown in the dialog. LN is the name of the process group that the object belongs to. The individual process objects are identified by indices [8,34].

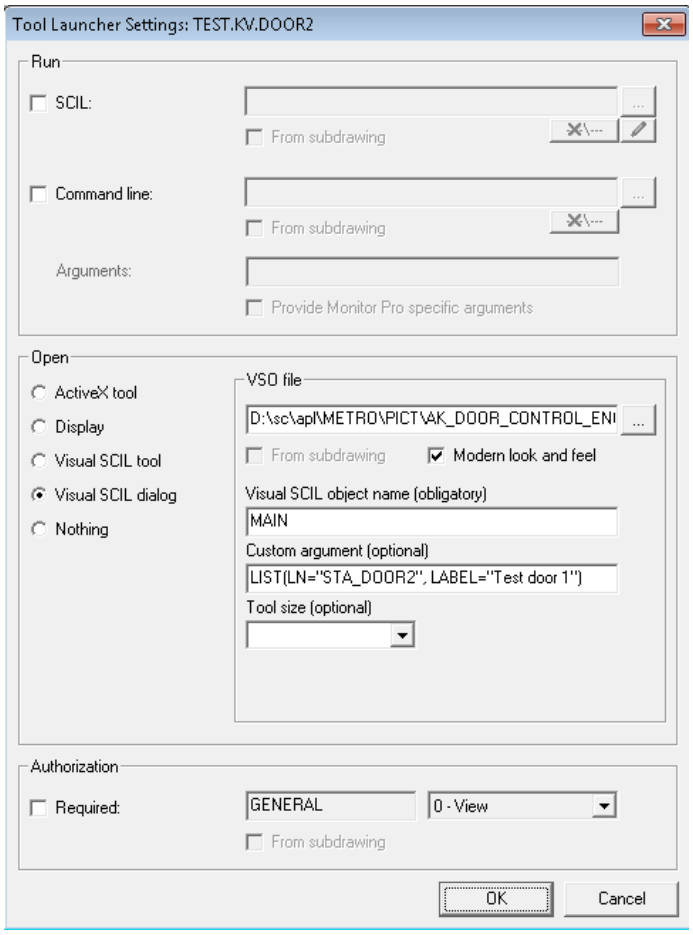

Figure 2. Tool Launcher settings for a door control dialog.

Almost all control dialogs made for the special functions of metro and railway applications have the same kind of appearance. Figure 3 shows the control dialog for one door type. State of the object and possible control buttons are located in the left side of the dialog. The right side of the dialog contains additional options, such as blockings and forced control button. If the object can be controlled by a time program, the settings for the time program are shown at the bottom of the dialog.

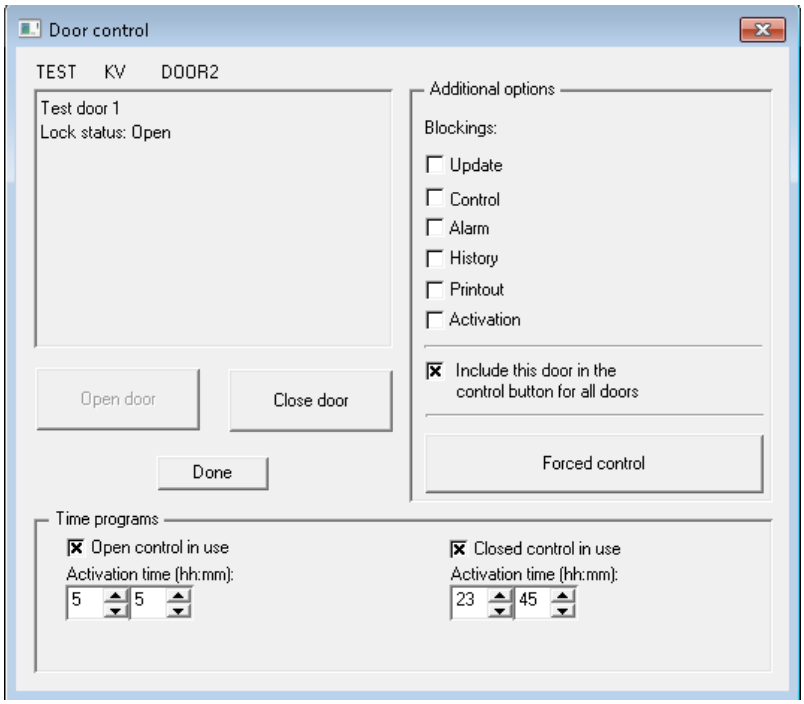

Figure 3. A control dialog made for one door type.

2.4 Creation of Subdrawings

Subdrawings are used when the same device symbol is needed multiple times in the same application. In MicroSCADA, subdrawings are used to represent states of the objects. The appearance and behavior of the subdrawings are defined by their data variables and custom attributes [14,61]. MicroSCADA includes a set of subdrawings for the most used devices in substation and process industry applications, for example breakers, generators and pipelines. It is possible to create new subdrawings and edit existing ones in the Display Builder.

Subdrawings are basically created the same way as process displays and they both have the same features. It is possible to draw different kind of graphical objects and add dynamic components to them. The following dynamic components can be set to a graphical object [14,207]:

- Attribute Dynamics
- Motion Dynamics
- Subdrawing Dynamics
- Visibility Dynamics
- Text Dynamics
- Blinking Dynamics

Dynamic components need data from the process to be able to show changes in it. That is why every dynamic component of the subdrawing is given its own data variable from one of the existing data sources. [14,211.] The data variables are mapped later to the right object when using the subdrawing in process displays. It is important to define the data variables of subdrawings as public when creating a subdrawing. Otherwise they will not show at the variable mapping list when using the created subdrawing as a symbol in a process display.

When a subdrawing is added to a process display, all data sources and dynamics from the subdrawing's original file are included in the subdrawing. It is possible to change the data the subdrawing uses by changing its mapped variables. [14,204.] Variable mapping is a useful feature because it offers a fast way to add variables of the right field device for each subdrawing. It also enables the user to use the same subdrawing for displaying different kinds of data and devices.

Figure 4 shows the data variables of an escalator subdrawing. It is possible to change the mapping type of subdrawing's data variables. In Figure 4, every data variable of the subdrawing is mapped to a variable from one of the data sources. The data variables can also be mapped to a constant value or left unmapped [14,204]. This enables the use of the same subdrawing even though all field devices do not have the same state indications.

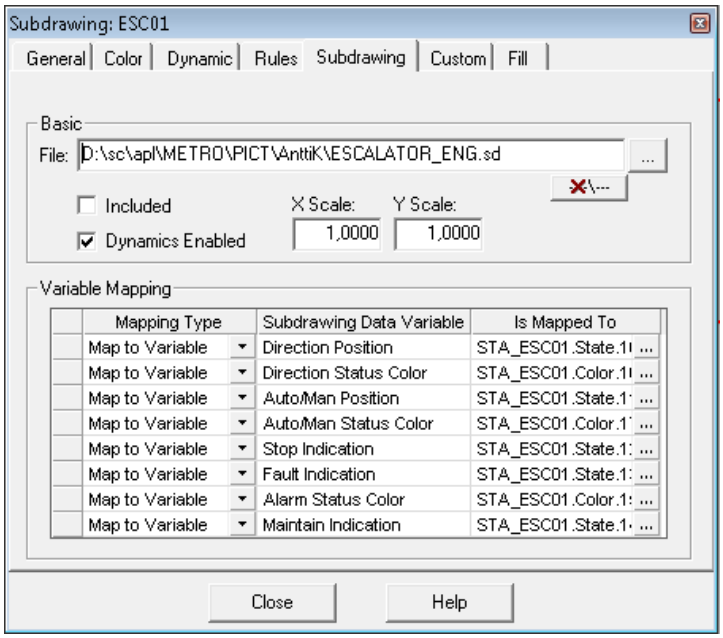

Figure 4. Data variables of a escalator subdrawing.

2.5 Process Objects in MicroSCADA

Most of the process objects in metro and railway applications are binary inputs and outputs. Binary inputs and binary outputs can have only two values, 0 and 1. The values from a binary input can be transferred into a double binary indication (DB) in the

MicroSCADA database, if more state information is needed for the object. DBs can have four different values: 0, 1, 2 and 3. Usually the added states are intermediate state and error state information, both of which are created by the statuses of other state inputs of the object with a command procedure. The DB process objects are usually not connected straight to the devices. Instead, they are used as auxiliary objects in the MicroSCADA database.

In addition to the binary I/O, analog I/O and pulse counters are also used. Analog inputs are used for getting measurement data from the process, for example temperature and brightness values. Analog outputs are used when a limit is set from the control system to a measurement in the process. MicroSCADA also has few other predefined process object types: digital input/output and bit stream. The programmer can also create their own process object types if needed. [15,44-45.]

Some PLCs in metro applications use analog inputs to transfer all states of one device to MicroSCADA. The analog input contains all states of one field device as different integer values. The integer values from the analog input are transferred into binary inputs in the MicroSCADA database. This is done, because the states of the field devices are easier to handle in a binary input form. For example, subdrawing mapping and control dialog programming is simpler, when the object values can only be 0 or 1. The usage of the analog inputs also significantly reduces the amount of process objects connected to PLCs.

The transfer from integer values into binary inputs is made by adding an event channel with a command procedure to the analog input in the MicroSCADA database. Event channels are used for the automatic start-up of event-bound activities [15,51]. A command procedure contains a SCIL program, which in this case transfers the integer

value into binary inputs. Example code in Listing 1 shows the command procedure needed for transferring integer state values of one door type into binary inputs.

```
#local val, b0, b1, b13
#error ignore

#if %os<>10 #then #block   ;checks object status
val = bin(trunc(%ov))      ;object value is truncated into integer and
                           ;turned into binary text

;creating a substring from specific parts of the binary text and
;converting it to integer
   b0 = dec_scan(substr(val, 32, 1))
   b1 = dec_scan(substr(val, 31, 1))
   b13 = dec_scan(substr(val, 19, 1))

;transferring values into binary inputs in MicroSCADA database
   #set 'ln'_H:pov10 = b13      ;Door open, alarm
   #set 'ln':pov8 = b0          ;Door position
   #set 'ln':pov9 = b1          ;State of supervision
#block_end
```

Listing 1. A command procedure for moving the object states from analog input into separate binary inputs.

The command procedure starts with declaring the variables used in it. After that it checks the Object Status (OS) value of the analog input. Object Status indicates the reliability of the registered data. If OS is 10, it means that the object has no value and, in this case, the command procedure will not continue. [15,48.] After that the command procedure truncates the Object Value (OV) and turns it into a 32-bit binary format with the BIN command. The 32-bit binary format is made up of 32 bits where each bit can have two possible states, 0 and 1 [16].

Next step is to create one-bit long substrings from specific parts of the binary representation of the integer values and then convert the substrings back to integer. Last

step is to move the integer values into OV's of pre-made binary inputs in the MicroSCADA database.

The command procedures for different field devices have a few differences. The amount of integer values depends on how many state inputs the field device has. The substring starting points also depend on what values the PLC is programmed to send to MicroSCADA. If the binary input values are transferred into a DB data type in the MicroSCADA database, there can also be some conditions for its values at the end of the command procedure.

All status changes and controls must be displayed correctly in the MicroSCADA Event and Alarm Lists. To do this, event handling objects are used. Every process object has an event handling object in it. Event handling objects include translatable texts that describe the state of the process objects. The event handling objects in MicroSCADA includes the most common state texts by default, but a lot of new event handling objects are needed for some metro and railway devices to correctly describe their operation. Figure 5 shows the event handling object for the escalator direction process object.

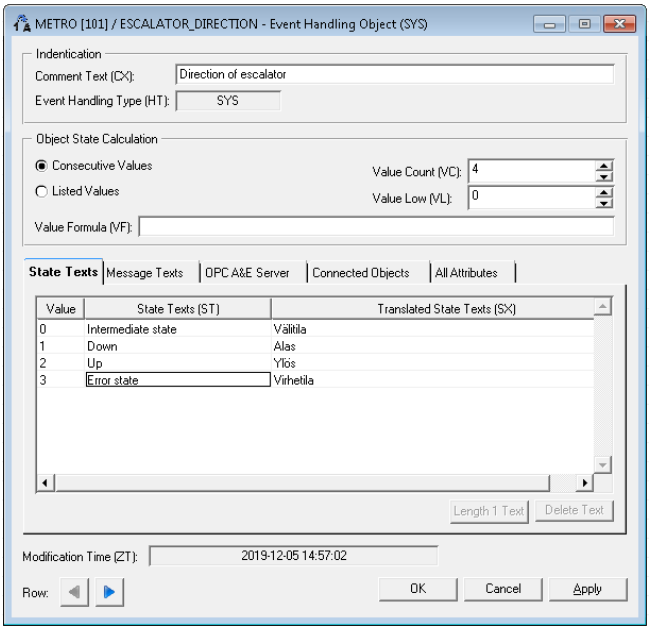

Figure 5. Event handling object for escalator direction states.

3 Special Functions in Metro and Railway Applications

MicroSCADA is used to supervise and control the infrastructure of metro stations and tunnels in current metro applications. For example, lighting, surveillance and smoke removal systems are included in these applications. The current MicroSCADA railway applications are, however, used to monitor and control the power supply of the railway network. The most used communication protocol in current metro and railway applications is IEC 60870-5-104.

In metro applications, device blocks are used to help the creation of databases in MicroSCADA and the programming of the PLCs. The use of device blocks also facilitates the overall planning of the metro system functionality, since all devices are organized into clear entities. All devices in one device block have the same functionality and

process objects. If device blocks are used, an auxiliary process object with the device block name in its Object Text (OX) attribute is added to every device in the MicroSCADA database. MicroSCADA identifies the device type from this OX attribute and changes the content of the control dialogs based on it, for example.

Metro applications require much more special functions than railway applications, because MicroSCADA is not designed for building automation. The railway applications do not have as many special functions as metro applications, because most of the needed functionality is already available in MicroSCADA. The special functions of both application types are documented below.

3.1 Rail Heating

Rail heating is implemented to keep the train tracks clear of snow and ice. In the old days, the rails were kept clear by using road salt and kerosene [17]. Nowadays an electric heating system is used. The electrical energy for the rail heating is taken through a transformer which converts the high voltage used in the rail system into a low voltage that is suitable for the heating equipment [18].

The rail heating can be monitored and controlled through MicroSCADA. There are slight differences between different heating types. In current railway applications, the following heating types are being used:

- Heating units with on/off control and enhanced heating mode
- Heating units with multiple heating modes
- Heating units that are divided into two or more groups

The databases for all heating units are quite similar. The only difference is the naming and number of the PLC signals. Figure 6 shows all process objects of the heating unit with multiple heating modes. The indices 11 and 100 are used as auxiliary inputs.

LN	IX	[UN]	[OA]/IN	OI		OX	SS	OV
TEST_HEATING1	1			TEST	HEATING1	Heating off control	1	1
TEST_HEATING1	2			TEST	HEATING1	Heating 1/2 control	1	1
TEST_HEATING1	3			TEST	HEATING1	Heating 3/4 control	1	1
TEST_HEATING1	4			TEST	HEATING1	Heating 1/1 control	1	1
TEST_HEATING1	10			TEST	HEATING1	No controls in memory -> controls object	1	0
TEST_HEATING1	11			TEST	HEATING1	Station connection	1	1
TEST_HEATING1	100			TEST	HEATING1	State of rail heating	1	4.000
TEST_HEATING1	101			TEST	HEATING1	Heating 1/2 state	1	0
TEST_HEATING1	102			TEST	HEATING1	Heating 3/4 state	1	0
TEST_HEATING1	103			TEST	HEATING1	Heating 1/1 state	1	0
TEST_HEATING1	104			TEST	HEATING1	Heating on state	1	0
TEST_HEATING1	1011			TEST	HEATING1	Station connection	1	1.000

Figure 6. Process objects of rail heating type with multiple heating modes in Object Navigator.

The rail heating presentation in the process displays consists of several different subdrawings. The base subdrawing includes the name of the heating unit and its state indications. It also has an alarm text on the bottom right corner that warns of the lack of control command in the memory. The line indicator subdrawing is added next to the unit's name to indicate the powered state of the unit. Control button that is configured to open the control dialog is added under the state indication field. Slightly modified "IED" subdrawing from the symbol library is also added to the right side of the base subdrawing. It indicates the connection to the unit with color and opens a control dialog which shows the status of the connection. The complete appearance of a rail heating unit with multiple heating modes is shown in Figure 7. Every heating unit type has its own base subdrawing with the correct state indications.

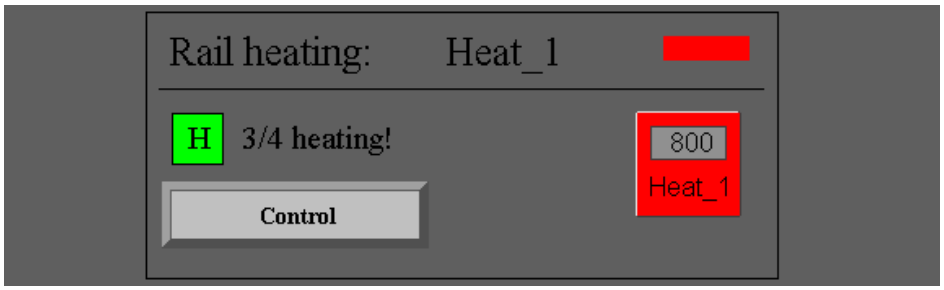

Figure 7. Indication of a rail heating unit in a Monitor Pro display.

A control dialog has been created for every rail heating type. Their appearance is kept as simple as possible. They show the name the and state of the unit and have control buttons for available heating modes. The control dialog for heating units with two control groups is shown in Figure 8.

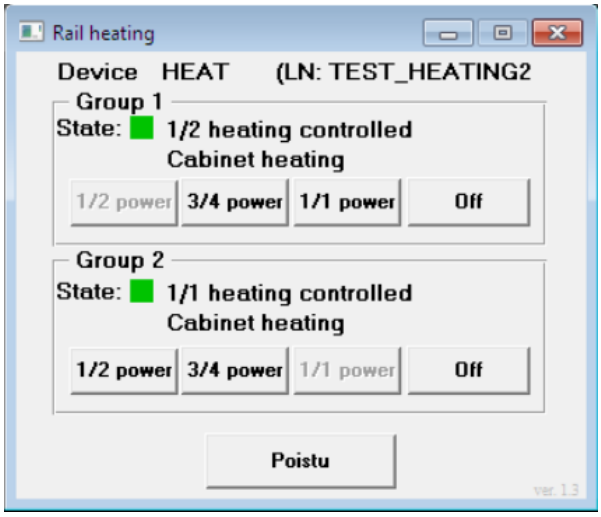

Figure 8. Control dialog of a heating unit with two control groups.

3.2 Access Control

Access control means monitoring and controlling the access to different parts of the supervised area. For example, it is possible to see if the door is open or closed and control the locks of them. In electronic access control it is also possible to use credentials to gain access to certain areas. A working access control prevents trespassing and makes stations effortless to use and supervise.

Access control handling is implemented in current MicroSCADA metro applications. Only the door monitoring and controlling is handled in MicroSCADA. Rest of the access control is handled in other systems. Because several types of doors are used in different metro stations, MicroSCADA needs to handle a lot of different PLC signals. All current inputs and outputs for doors are listed in Table 1.

Table 1. PLC signals for doors in MicroSCADA database.

Inputs	Outputs
Lock status	Lock door
Door position	Unlock door
State of door supervision	Open door
Motion detector block	Close door
Door open (alarm)	Supervision on
Control conflict of door position (alarm)	Supervision off
Control conflict of door lock (alarm)	
Status change of key switch	

PLCs at the stations can be programmed to send the status information of doors to MicroSCADA as separate binary inputs or as one analog input which contains all inputs

as different values. If analog input is used, the values are transferred into binary inputs in MicroSCADA database. Door position values are transferred into a DB from the analog input to enable intermediate and error state indications to them. Controls commands are sent to PLC as binary outputs. Also, new event handling objects have been created for the inputs and outputs of doors so that their states are shown correctly in Event and Alarm Lists.

MicroSCADA has a door symbol in its symbol library but in metro applications it is replaced with a new subdrawing. The original door symbol does not have enough state indications and does not fit to the design of the metro application process displays. A new control dialog for doors has also been created.

Control dialogs for doors includes the following features:

- Name and description of the door
- Object state
- Lock or supervision state
- Door position
- Blockings
- Alarms
- Lock or supervision control
- Door position control
- Forced control
- Option to control the door from group control dialog
- Time program for locking or supervision

The features shown on the dialog depend on the door type. Different door types require different types of control and status information to be shown on the dialog. Also, the

appearance and content of the control dialog is changed depending on which process object indices are used for the specific door.

Three different subdrawings for doors have been created to be used in the process displays. All of them suit a specific door type. The simplest subdrawing includes three variable mappings: position, status color and star indication. Star indication is shown when the door is included to be controlled from the group control dialog. Second subdrawing includes motion detector block and key switch indications in addition to indications above. Third subdrawing is based on the second one and is meant for a listed view of door groups in process displays. It has an object name mapping, which is located next to the subdrawing. All three subdrawings are shown in Figure 9.

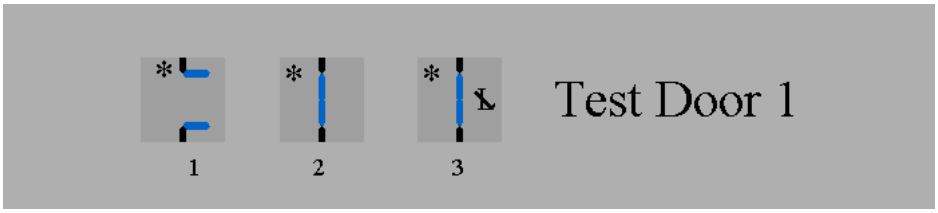

Figure 9. Subdrawings for different door types in a Monitor Pro process display.

It is possible to map different states into the position mapping of the door subdrawings. Depending on the door, the position mapping can indicate the door position, lock or supervision states. This allows the use of the same subdrawing for different door state indications.

3.3 Surveillance

Surveillance is an essential part of the security of metro stations. The surveillance of the station is usually done with passive infrared (PIR) motion sensors. PIR sensors detect

changes in the infrared radiation from the environment. Brightness of the surveillance area does not affect the operation of the PIR sensor. [19.] The sensors are placed across the station to get all important areas under surveillance. When a sensor detects movement, it generates an electrical signal. This signal is used as an alarm in MicroSCADA.

The sensors in the public areas need to be active only when the station is closed. Otherwise they would send alarms all the time when people are walking in the station. To prevent this, the sensors can be blocked from the station's PLC. The blocking controls and states are available in MicroSCADA so the operator can control the state of the sensors. It is also possible to use a time program to switch surveillance on and off automatically in obedience to the opening hours of the station.

New subdrawings and control dialogs have been created for the sensors. A regular Alarm Led subdrawing is used for the alarms, but the blocking indication is tailor-made. Control dialog for the PIR sensors is very similar to the door control dialog. It shows the state of sensor blocking, active alarms, time programs and has control buttons for blocking. It is also possible to change the name of the sensor from the control dialog.

To show the blocking state and the execution times of time program in a process display, a new subdrawing is created. Data variables of the subdrawing are shown on Figure 10. Figure 10 also shows the appearance of the PIR sensor subdrawing in a process display. The alarm indication of the sensor is added next to the subdrawing. The upper sensor has an active time program and is blocked between 04.45 – 00.00. The lower sensor has no active time program and is not blocked. The lower sensor is also in alarming state.

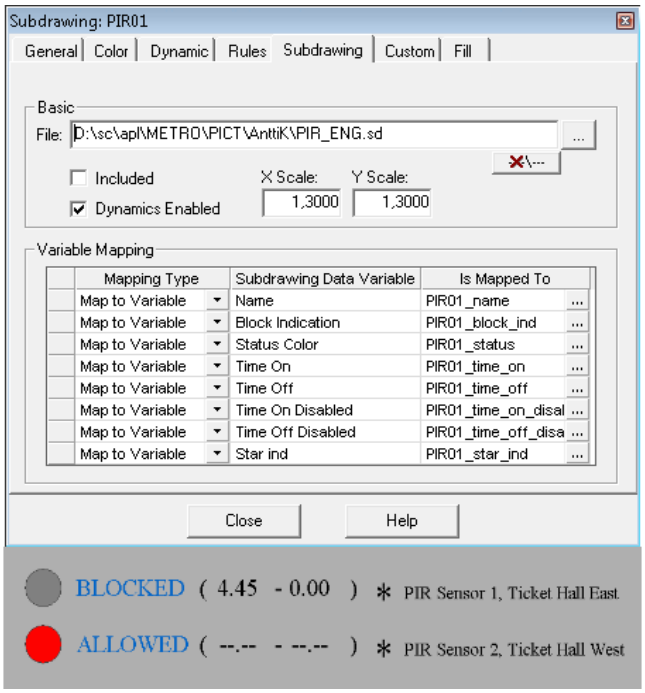

Figure 10. Data variables of a PIR sensor subdrawing and its appearance in a process display.

3.4 Escalators and Elevators

Escalators and elevators are important for the accessibility of metro stations because most of them are located underground. Long escalators and many elevators are needed for a smooth movement of passengers from street level to stations. For example, the longest escalator in Finland is located in Finnoo metro station, which is still under construction. It is approximately 78 meters long and its lifting height is 33,5 meters [20].

In metro applications, escalators can be monitored and controlled through MicroSCADA. Usually only the monitoring is enabled because it is not safe to control escalators remotely from the SCADA system. There might be people using the escalator when the

operator from the control room decides to control it. The escalators are also programmed to automatically start running towards the exits of the station when the fire alarm is activated.

Depending on the station, MicroSCADA receives the state inputs of escalators from the station's PLC as separate binary inputs or as one analog input which contains all states of the escalator. All escalator inputs and outputs used in MicroSCADA database are listed in Table 2. Direction and running state values are moved into a double indication input to enable intermediate and error state indications to them. To show the states and controls of the escalators correctly in Event and Alarm Lists, new event handling objects have been created for the escalator inputs and outputs.

Table 2. Inputs and outputs used for escalators in MicroSCADA database.

Inputs	Outputs
Escalator not running	Automatic operation down
Direction of escalator (DB)	Automatic operation up
Running state of escalator (DB)	Manual operation up
Escalator in service	Manual operation down
Technical fault (alarm)	Stop escalator
Emergency stop pressed (alarm)	
Escalator stopped (alarm)	

Automatic operation means that the escalator starts moving only when people step into it. In manual operation mode the escalator is running non-stop.

MicroSCADA symbol library includes escalator subdrawings, but their appearance and the lack of data variables means that they are not suitable for the existing metro

applications. Because of that, new subdrawings and control dialogs have been created. Control dialogs for escalators includes the following features:

- Object status
- Direction of the escalator
- Running type (automatic/manual)
- Alarms
- Blockings
- Control buttons (if used)

Subdrawings for escalators have been designed to show all necessary indications clearly to the operator. The operator can quickly check the statuses of all escalators of the station straight from the subdrawings without opening the control dialog. Figure 11 demonstrates the appearance of the escalator subdrawings in a process display. Escalator number one is not running and is in intermediate state. Technical fault and alarm indications are also on. Escalator number two is running in automatic mode and its direction is down.

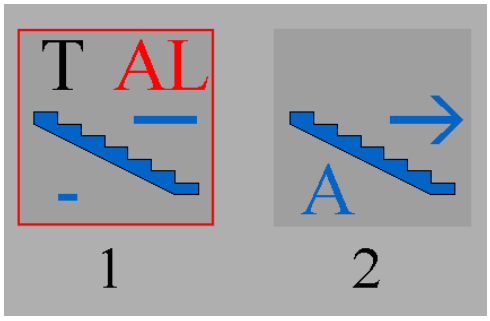

Figure 11. Two escalator subdrawings in a Monitor Pro process display.

Elevators require much less engineering work in MicroSCADA than escalators. Only the alarms of the elevators are monitored through MicroSCADA. MicroSCADA symbol library includes an alarm symbol, so no special functions are needed for alarm handling.

Only special function for the elevators is the making of a simple base subdrawing for the elevator alarm indication. The subdrawing has only two mappings: maintain and status color indications. The Alarm Led subdrawing is added on top of it to indicate the alarms of the elevator. The maintain state of the elevator can be handled as an alarm or a regular binary input in MicroSCADA. Figure 12 shows the elevator subdrawing with the alarm indication in a process display. The maintain indication is on and the subdrawing has a blue frame. Blue color means in this example application that the value of the mapped status variable is simulated in MicroSCADA. Simulated value means that the OV of the process object is manually entered in MicroSCADA database instead of receiving it from the process.

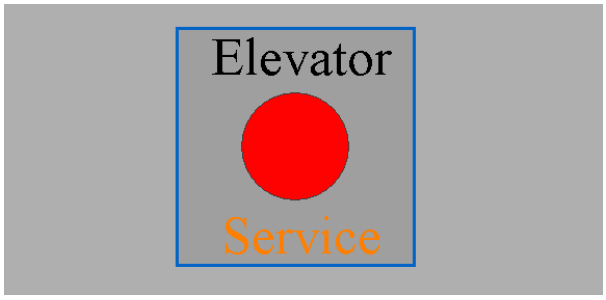

Figure 12. Elevator subdrawing and alarm indication in a Monitor Pro process display.

3.5 Lighting

The lighting of the railway or metro station can be monitored and controlled from MicroSCADA. The tunnel lighting between the metro stations can also be included to the SCADA system. With time programs, MicroSCADA is able to automatically turn off the lights when the station is closed and turn them back on when the station is about to open.

Lighting solutions of the stations depend on their age. Old stations have standard relay-controlled lighting system and newer stations may use modern control systems such as DALI (Digital Addressable Lighting Interface). DALI is an open interface lighting control system which can be set to communicate with PLCs [21].

Different lighting solutions mean that a lot of different PLC signals are needed to be handled in MicroSCADA. All PLC signals used for lighting are shown in Table 3. New event handling objects are also created for the process objects to make the states of them appear correctly in the Event List. Lights can be controlled individually or only as groups depending on the station. It is possible to create group controls for individually controlled lights as well in MicroSCADA. The lights can also be controlled by lux limits and twilight switches. States and limits of them can be monitored and changed from MicroSCADA. Rest of the lux limit and twilight switch handling is done in the PLCs of the stations.

Table 3. PLC signals used for lighting in MicroSCADA.

Inputs	Outputs
Lighting state on/off	Lighting on
Control state on/off	Lighting off
DALI control cabinet in use	Lux limit on
Control conflict	Lux limit off

Lux measurement	Lux limit writing to PLC
Lux limit in use	Twilight switch on
Lux limit reached	Twilight switch off
Twilight switch state on/off	

The handling of the lighting in process displays requires new control dialogs and subdrawings. Because the lighting solutions of the stations varies significantly, multiple dialogs and subdrawings have been created to cover them all.

Subdrawings for lights are designed to look simple and easy to understand from the operator's perspective. In normal lights, the bulb figure shows the state of the light and the text under it indicates the state of control. If the control state input is not available, the text is mapped to indicate the state of the light. The subdrawing for the DALI-controlled lights is a bit different. The On/Off text has been changed to Auto/Man and is mapped to the light state indication. Control off state is indicated with a large cross over the subdrawing and the bulb figure is mapped to indicate the state of the control cabinet.

Figure 13 shows two different types of group-controlled lights in a process display. The left light group is DALI-controlled, and it has been controlled off. The right group is a normal group-controlled light group which has been controlled on. The subdrawings appear as pink because the object statuses of the lights are uncertain. It means that the connection to the station is lost.

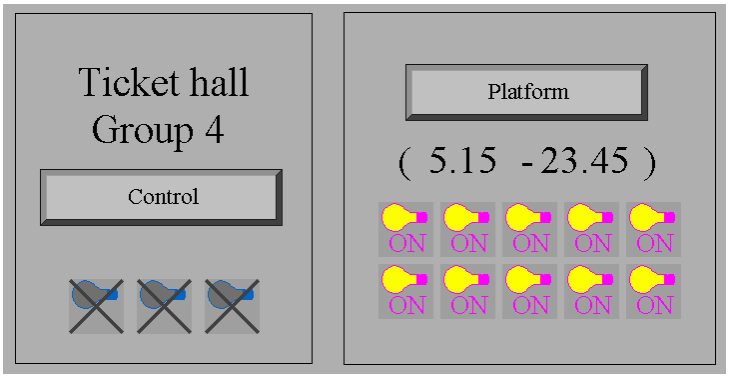

Figure 13. Group-controlled lights in a Monitor Pro process display.

The appearance of the control dialogs for lights is quite similar on every light type. They all include the following elements:

- Status and object state indications on the text field
- On/off control
- Forced control
- Blockings
- Time program settings

In addition, control dialog for DALI-controlled lights has a button to force the lights 100% on.

3.6 Ventilation and Smoke Removal

Ventilation and smoke removal systems are important parts of the functionality and security of the metro stations. Ventilation system is designed for maintaining good air quality in the station: it provides fresh air for breathing and removes used carbon dioxide

saturated air [22]. The purpose of a smoke removal system is to secure people's escape from the station by keeping the smoke out of the escape routes. It also helps the rescue and extinguishing work as well as to reduce the damage caused by the smoke. [23.] Because most of the metro stations are located underground, both systems must operate perfectly especially in case of emergency.

Ventilation and smoke removal in metro stations is done by different fans, flaps, dampers and air grilles. They all require several engineering phases to make them function correctly in MicroSCADA. The age of the station must also be considered in MicroSCADA database engineering. The ventilation and smoke removal systems in new stations are much more large-scaled and diverse than the systems in older stations. This has occurred because the device development has been fast and the regulations considering safety have become stricter. The larger systems also mean, that the amount of PLC signals coming from the modern systems is huge compared to older systems.

3.6.1 Ventilation and Smoke Exhaust Fans

The purpose of the ventilation and smoke exhaust fans is to provide fresh air to the station and to remove smoke from it. The fans used in the metro stations have big differences between them. Some fans can be controlled in and out, some have different speed settings, and some are controlled through a frequency converter which enables precise fan control through a control circuit. MicroSCADA needs to handle all these fan types. It means that a lot of new subdrawings and control dialogs are needed to be made.

The process objects in MicroSCADA database vary greatly depending on the type of fan. Fans in the older metro stations are usually much simpler than in the new ones. They use binary inputs and outputs for fan states and controls. The inputs and outputs for fans in modern stations are much more complicated. They feature much more state inputs

and alarms as well as many analog inputs and outputs for the control circuits of the fans. Because of the big differences of the fans, the old and new fans will be documented separately.

Older Fans

The inputs and outputs used in older fan types are listed in Table 4. Used inputs and outputs for each fan depend on their features and intended use. A double indication auxiliary state input is also added for each fan in the MicroSCADA database. In some cases, the auxiliary input's data type can also be analog input if the fan can generate more than four states. The auxiliary state input makes event handling and variable mapping of the subdrawing easier because it includes all states of the fan in one process object.

Table 4. PLC signals of older fan types in MicroSCADA database.

Inputs	Outputs
Control switch status (local/remote)	Fan in
Fan blowing in	Fan out
Fan blowing out	Stop fan
Fan stopped	Fan 1/1 speed
Fan speed 1/1	Fan ½ speed
Fan speed ½	
Thermal overload relay (alarm)	

All older fans use the same subdrawing base. The difference in each subdrawing is in the direction and speed indications. Because of the auxiliary state input in the database, the fan subdrawings have only two data variables in them. It means that variable

mapping for each fan is fast and simple. Figure 14 shows two fans in a process display. The fan on the left side is blowing out in 50 % speed. The fan in the right side is blowing in in full speed.

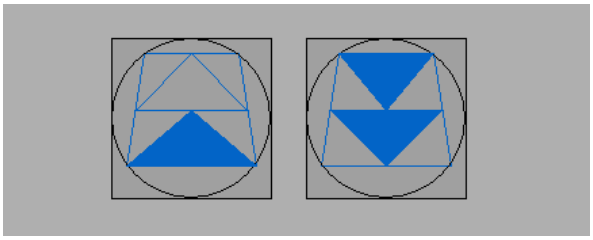

Figure 14. Two fan subdrawings in a Monitor Pro process display.

Two different control dialogs have been created for older fan types. The biggest difference in them are the control buttons. One control dialog is made only for changing the direction of the fan and the other is able to change the fan speed as well. Figure 15 shows the Finnish version of the control dialog for fans with direction and speed control. The direction buttons are programmed to be dimmed when the fan is running. The direction buttons become usable by stopping the fan first.

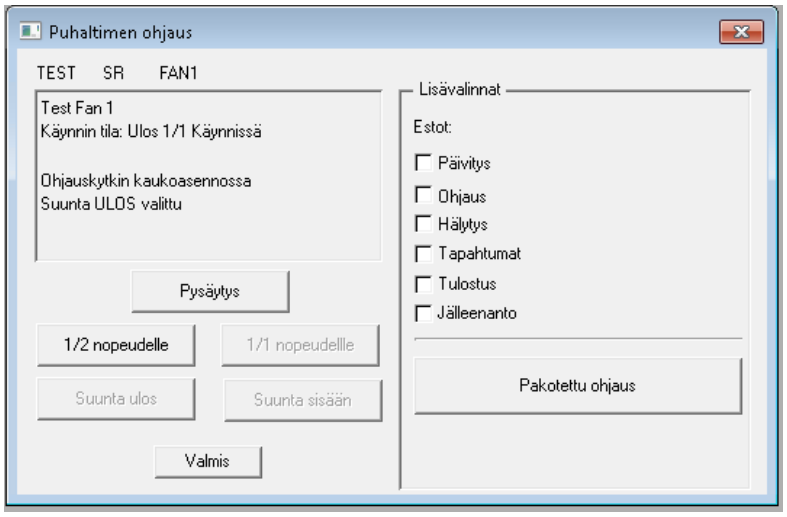

Figure 15. Control dialog for fans with direction and speed control.

Modern Fans

Many fans in newer metro stations are controlled through a frequency converter. Besides these there are also fans which have similar control options as older fan types. The frequency converter has a control circuit from which the fans can be precisely controlled. These control circuits can be adjusted from MicroSCADA. Because of the control circuits, the amount of process objects is notably larger than in older fans. Modern fans also have more state inputs and alarms, but the main difference compared to older fans is the use of the control circuit. If the fan is controlled from multiple PLCs, the number of process objects is increased even more.

Figure 16 shows the process objects of one fan which has a control circuit in the Object Navigator. This fan is controlled from two safety PLCs. Having duplicated control ensures that the device will operate if one PLC fails.

LN	IX	[UN]	[OA]/IN	[OB]/EH	OI	OX	OV
STA_FAN4	1				TEST SR FAN4	SB_SP_PUHALLIN_SC	0
STA_FAN4	3				TEST SR FAN4	Safeety PLC2 active	0
STA_FAN4	4				TEST SR FAN4	Controlled on	0
STA_FAN4	5				TEST SR FAN4	State of control circuit	2
STA_FAN4	6				TEST SR FAN4	Control circuit forced manual from panel	1
STA_FAN4	7				TEST SR FAN4	Locking OK	0
STA_FAN4	8				TEST SR FAN4	Running permission	0
STA_FAN4	9				TEST SR FAN4	Running state	1
STA_FAN4	50				TEST SR FAN4	On control PLC1	1
STA_FAN4	51				TEST SR FAN4	Stop control PLC1	1
STA_FAN4	53				TEST SR FAN4	On control PLC2	1
STA_FAN4	54				TEST SR FAN4	Stop control PLC2	1
STA_FAN4	60				TEST SR FAN4	Control circuit control PLC1	1
STA_FAN4	61				TEST SR FAN4	CC auto state pressure write PLC1	306
STA_FAN4	62				TEST SR FAN4	CC man state pressure write PLC1	57
STA_FAN4	63				TEST SR FAN4	Control circuit control PLC2	1
STA_FAN4	64				TEST SR FAN4	CC auto state pressure write PLC2	310
STA_FAN4	65				TEST SR FAN4	CC man state pressure write PLC2	57
STA_FAN4	70				TEST SR FAN4	CC fan speed PLC1	1300.0
STA_FAN4	71				TEST SR FAN4	CC manual state speed PLC1	30.0
STA_FAN4	72				TEST SR FAN4	CC auto state pressure PLC2	21.5
STA_FAN4	73				TEST SR FAN4	CC fan speed PLC2	350.0
STA_FAN4	74				TEST SR FAN4	CC manual state speed PLC2	6.0
STA_FAN4	75				TEST SR FAN4	CC auto state pressure PLC2	21.0
STA_FAN4	170				TEST SR FAN4	CC fan speed over or under the limit PLC1	0
STA_FAN4	173				TEST SR FAN4	CC fan speed over or under the limit PLC2	0
STA_FAN4	270				TEST SR FAN4	CC fan speed upper limit alarm PLC1	0
STA_FAN4	273				TEST SR FAN4	CC fan speed upper limit alarm PLC2	0
STA_FAN4	370				TEST SR FAN4	CC fan speed lower limit alarm PLC1	0
STA_FAN4	373				TEST SR FAN4	CC fan speed lower limit alarm PLC2	0
STA_FAN4	996				TEST SR FAN4	CC Analog input reading PLC1	2.000
STA_FAN4	997				TEST SR FAN4	CC Analog input reading PLC2	0.000
STA_FAN4	998				TEST SR FAN4	Analog input reading PLC2	0.000
STA_FAN4	999				TEST SR FAN4	Analog input reading PLC1	0.000
STA_FAN4_AL	10				TEST SR FAN4	Running permission or running fault	0
STA_FAN4_AL	11				TEST SR FAN4	Safety switch or control voltage	0
STA_FAN4_AL	12				TEST SR FAN4	Locking on	0
STA_FAN4_AL	13				TEST SR FAN4	Frequency converter fault	0
STA_FAN4_AL	14				TEST SR FAN4	Fire mode activated	0
STA_FAN4_AL	15				TEST SR FAN4	Starting step 1 or 2 fault	0
STA_FAN4_AL	16				TEST SR FAN4	Control conflict	0
STA_FAN4_AL	17				TEST SR FAN4	Pressure difference measurement	0
STA_FAN4_AL	18				TEST SR FAN4	Hälytysindikointiestot	0.000

Figure 16. Process objects of a fan equipped with a control circuit in the Object Navigator.

The modern fans are usually a part of a smoke removal block. Subdrawings used in smoke removal block process displays will be handled later. The fans in the pressure equalization gaps of the metro tunnel, however, are visible in main smoke removal process displays. The appearance of these subdrawings is very simple: it shows the direction and the status of the fan with circles and arrows.

The same control dialog file is used for every modern fan. The appearance and features of the dialog depend on the device block of the fan. The control dialog is programmed to check the fan's device block. Based on this check, certain parts of the control dialog become visible and usable.

3.6.2 Smoke Dampers and Air Grilles

Smoke dampers play an important role in the smoke removal systems of metro stations. They prevent the smoke from spreading to other areas of the station. It is possible to use a combination of fans and dampers to create an overpressure into a certain area of the station to exhaust smoke from it [24]. Smoke exhaust flaps are used on facilities which are located at ground level.

Air grilles are located in the pressure equalization gaps of the metro tunnels. They are used to control the temperature of the metro tunnel. [25.] Air grilles are motorized which means that they can be controlled from MicroSCADA. Controlling of air grilles can be manual or automatic by thermostat.

Because the operating principle of these devices is very similar, their PLC signals are also almost identical. As with fans, the age of the system affects the data coming from these devices. The differences are however smaller than with the fans which means that all these devices can be processed at the same time.

Figure 17 shows the process objects of an older air grill and a modern smoke damper. The control outputs are similar in both devices, but the smoke damper has more comprehensive state inputs. The smoke damper uses less IEC addresses despite the larger amount of state inputs, because MicroSCADA receives all its states from the PLC in one analog input.

LN	IX	[UN]	[OA]/IN	[OB]/EH	OI	OX	OV	SS
STA_AIRGRILLE1	10				TEST SR AIRGRILLE1	Object status	1	1
STA_AIRGRILLE1	11	101	3000		TEST SR AIRGRILLE1	State of control switch	0	1
STA_AIRGRILLE1	12	101	3001		TEST SR AIRGRILLE1	Grill closed	0	1
STA_AIRGRILLE1	13	101	3002		TEST SR AIRGRILLE1	Grill open	1	1
STA_AIRGRILLE1	15	101	4000		TEST SR AIRGRILLE1	Close control	0	1
STA_AIRGRILLE1	16	101	4001		TEST SR AIRGRILLE1	Open control	1	1

LN	IX	[UN]	[OA]/IN	[OB]/EH	OI	OX	OV	SS
STA_DAMPER1	1				TEST SR DAMPER1	SMOKE_EXHAUST_DAMPER_1	0	1
STA_DAMPER1	2				TEST SR DAMPER1	Operating direction	1	1
STA_DAMPER1	3				TEST SR DAMPER1	Safety PLC active	0	1
STA_DAMPER1	6				TEST SR DAMPER1	Controlled Closed	0	1
STA_DAMPER1	7				TEST SR DAMPER1	Controlled Open	1	1
STA_DAMPER1	8				TEST SR DAMPER1	Damper status	2	1
STA_DAMPER1	9				TEST SR DAMPER1	Position status	2	1
STA_DAMPER1	50				TEST SR DAMPER1	Open control		0
STA_DAMPER1	51				TEST SR DAMPER1	Close control		0
STA_DAMPER1	53	102	4010		TEST SR DAMPER1	Open control Safety PLC	1	1
STA_DAMPER1	54	102	4011		TEST SR DAMPER1	Close control Safety PLC	1	1
STA_DAMPER1	998	102	3010		TEST SR DAMPER1	Status word reading Safety PLC	0.000	1
STA_DAMPER1	999				TEST SR DAMPER1	Status word reading		0
STA_DAMPER1_AL	10				TEST SR DAMPER1	Closed state not reached	0	1
STA_DAMPER1_AL	11				TEST SR DAMPER1	Open state not reached	0	1
STA_DAMPER1_AL	12				TEST SR DAMPER1	State conflict	0	1
STA_DAMPER1_AL	18				TEST SR DAMPER1	Alarm indicator blockings	0.000	1
STA_DAMPER1_AL	19				TEST SR DAMPER1	Alarm indicator selected on monitor	0.000	1

Figure 17. Process objects of an air grill and a smoke damper in the MicroSCADA database.

The same subdrawing is used for smoke dampers, smoke exhaust flaps and air grilles. They all function in a similar way so it is practical to use the same subdrawing for all of them. The direction of the line in the subdrawing indicates the state of the device. A simple circle-shaped subdrawing has also been created. It is used in the metro tunnel process displays where all smoke removal devices have a similar appearance. This simple subdrawing only shows the object status in the outer part and the door position in inner part of the circle with different colors. Both above subdrawings are displayed in Figure 18.

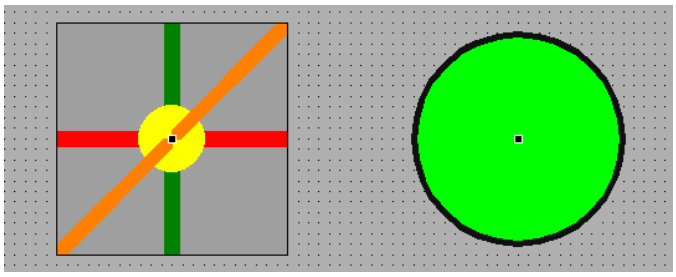

Figure 18. The subdrawings used for smoke dampers, smoke exhaust flaps and air grilles in the Display Editor.

Two different control dialogs have been created for the old and new devices. The main difference in them is that the control dialog for new devices displays more information about the object and it also includes the alarms generated by the device. Control dialogs for older devices do not have this feature, which means that the alarms are indicated with the Alarm Led subdrawing next to the subdrawing of the device.

3.6.3 Smoke Doors

Smoke doors are used in metro tunnels to prevent smoke from spreading across the metro tunnel in case of fire [26]. They can be controlled manually or automatically. The operation of doors is quite similar in old and new systems. Only difference is that the new ones provide more state inputs.

The PLC signals coming from smoke doors are listed in Table 5. The older door types have only the door position information, control switch states and some alarms. The door position information is transferred into a DB input in the MicroSCADA database to enable intermediate and error states as well as to make subdrawing mapping and event handling easier.

Table 5. Process objects of fire and smoke doors.

Inputs	Outputs
State of control switch (local/remote)	Close smoke door
Smoke door closed	Open smoke door
Smoke door open	Stop smoke door
Operating direction	
Safe automation active	
Control permit	
Controlled on	
Controlled off	
Locking OK	
State of lock	
Manual use	
MCB tripped (alarm)	
Closed state not reached (alarm)	
Open state not reached (alarm)	
Control conflict (alarm)	
Fault alarm	
Locking on (alarm)	
Blocking of use (alarm)	

The subdrawing for normal doors has been used as a base drawing for the smoke door subdrawing. It has position and status color indication data variables which means it is easy to map for all smoke doors of the system. The simple circle-shaped subdrawing presented in section 3.6.2 is also used for smoke doors in the metro tunnel process displays.

Two different control dialogs have been created for the older and newer smoke doors. They are almost identical with the control dialogs of smoke dampers, smoke exhaust flaps and air grilles. There are only minor changes in the dialog texts.

3.6.4 Smoke Removal Blocks

If a metro station has a large amount of different smoke removal devices, it is possible to use smoke removal blocks to control them. A smoke removal block (SRB) includes all smoke removal devices from a specific part of the metro station or tunnel. The usage of SRBs makes the smoke removal system faster and more efficient.

The SRBs are created in the station's PLC and MicroSCADA monitors and controls the states of the blocks. The process objects used for the SRBs are listed in Table 6. Most of the SRBs are connected to two different PLCs to ensure their operation in case of emergency.

Table 6. Process objects of smoke removal blocks used in MicroSCADA database.

Inputs	Outputs
Running state (DB)	Control on
Standby state	Control off
State of control switch	Control stop
Safety PLC 2 active	
Actuator fault (alarm)	
Control switch in stop state (alarm)	
Fire alarm test (alarm)	

Handling of the SRBs in the process displays requires new subdrawings and control dialogs. The SRB subdrawing is shown in Figure 19. On the left side of the figure is the base subdrawing which includes all state indications of the SRB. The buttons and additional text fields are added on top of the base subdrawing when creating the smoke removal process displays. The final version is shown on the right side of the figure.

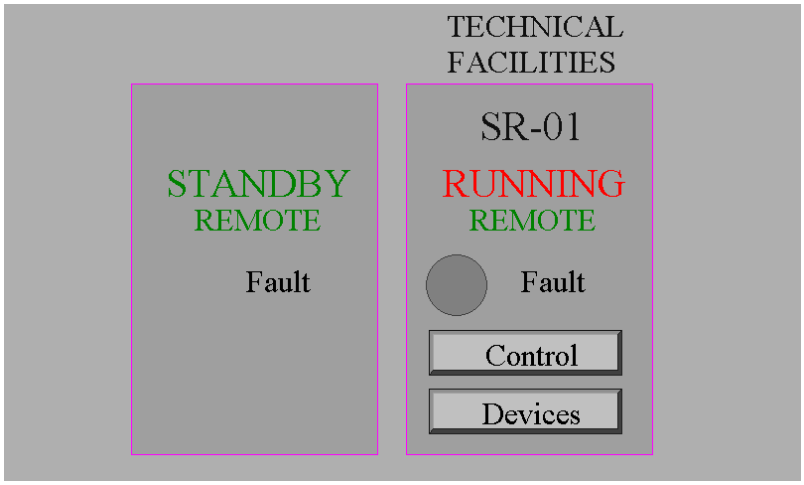

Figure 19. Base SRB subdrawing and the final version in a Monitor Pro process display.

Clicking the Control button opens the control dialog for the SRB. From the control dialog it is possible to see the state of the SRB and control it to run or stop. The Devices button opens a process display, which has all the devices of the SRB in a listed view. New subdrawings have been made to show the devices in the list. The appearance of the subdrawings is very similar to the PIR sensor subdrawing and it shows the name, state and alarms of every device in the SRB. For some fans, the pressure differential measurement is also included.

3.7 Fire Sprinkler System

The purpose of fire sprinkler systems is to help suppress and control fires in the facility. Fire sprinkler system starts to discharge water from the sprinklers when a specific temperature is reached. In addition to sprinklers themselves, water supply piping and control valves are also important components of a fire sprinkler system. [27.]

The metro networks are enabled with fire sprinkler systems. It is possible to supervise and control the sprinklers and valves of the fire sprinkler system from MicroSCADA. The PLC signals from the sprinklers and valves used in MicroSCADA are listed in Tables 7 and 8.

Table 7. Process objects of sprinklers in MicroSCADA database.

Inputs	Outputs
Fire alarm on	Manual trigger on
Automatic launch blocking on	Manual trigger off
Controlled on manually (alarm)	Automatic trigger blocking on
Automatic trigger blocked (alarm)	Automatic trigger blocking off

Table 8. Process objects of control valves in MicroSCADA database.

Inputs	Outputs
State of valve	Open Control
Position of valve (%)	Closed Control
Valve state over or under the limit	
Valve state high limit alarm	
Valve state low limit alarm	
State and control conflict (alarm)	
Electricity supply (alarm)	
Thermal overload relay (alarm)	
Valve stuck (alarm)	
Valve state closed (alarm)	

New subdrawing has been created for the sprinklers. Its appearance is very similar to the smoke removal block subdrawing. The base subdrawing shows the state of the sprinkler. Control dialog button and alarm indication is added on top of the base subdrawing. For valves, the “Gate Valve with Motor” subdrawing is used from the MicroSCADA Pipeline library.

Both sprinklers and valves also require new control dialogs. Control dialog for sprinklers shows the state and alarms of the sprinkler and it has buttons for automatic trigger blocking and manual triggering. The control dialog for valves is quite similar to the control dialog of smoke doors. The biggest difference is that the exact position of the valve as a percentage is added to the bottom right corner of the dialog.

3.8 Safe Automation Testing

All safety related parts of the metro systems are needed to be tested regularly to ensure their functionality. Most of these tests are related to the smoke removal and overpressurization systems. The test functions are performed in the PLCs of the stations, but they can be selected, started and monitored through MicroSCADA. The test intervals vary greatly depending on the device type. In current metro applications, the test intervals are between two days and one year and the tests are divided into the following groups:

- Smoke dampers
- Smoke exhaust flaps, doors and ventilation
- Automatic SRBs
- Manual SRBs of the station areas
- SRBs of pressure equalization gaps of the tunnel
- Linear heat detector relay cabinets
- Manual test for every group

Each station of the metro network has its own process objects for safe automation testing. The number of process objects per station is remarkably large, because every test block has a fault alarm input. In addition to these, a lot of state inputs about the state of the current test and the safe automation systems are needed. Several event handling objects have been created to display the test functions correctly in the Event and Alarm lists. Every test block is given its own integer value in the PLC which means, that the wanted test block is selected by sending its integer value as an analog input from MicroSCADA to PLC.

A process display has been created for safe automation testing from which it is possible to supervise and control the tests of the entire metro system. The display contains the following features:

- Test intervals for every group
- Time and date of the latest tests
- Indications for the need of testing
- Current state of testing
- A text field about the issues to be considered during testing
- Access to the control dialogs of safe automation tests for each station

New subdrawings are not needed for the safe automation testing. All indications are possible to create with the existing symbols. Instead, the control dialog has been created from scratch. The VSO-file of the dialog is divided into a main window and test elements for every test group. The main window includes the name, alarms and the current test of the station. The tests can be chosen and controlled from the test elements. They also include the latest execution date and the test interval of the selected test. It is also possible to disable the alarm for the need of test for each test group from the control dialog.

3.9 Trace Heating

During winter season, temperatures in metro tunnels can drop significantly. If the temperature goes sub-zero, the pipelines can freeze and burst because of ice expansion. That is why trace heating elements are needed in metro tunnels. The purpose of the trace heating is to maintain the wanted temperature in the piping of the process [28]. The trace heating cables can be self-regulated or controlled by a thermostat.

In current metro projects, trace heating is implemented with thermostat or season control. Season control means that the heating has summer and winter modes. It is possible to see the states of the heating elements and control them in MicroSCADA. The PLC signals used in MicroSCADA database for trace heating are shown in Table 9.

Table 9. PLC signals of trace heating in MicroSCADA database.

Inputs	Outputs
Locking state	Control on
Thermostat state	Control off
Control state	Thermostat enabled
Heating state	Thermostat bypass
Forced control state	Summer mode on
Control conflict (alarm)	Winter mode on
Thermal protection lock (alarm)	

In process displays, the trace heatings are shown in a listed view. Two new subdrawings have been created to display the state of the heatings in the list. Figure 20 shows two trace heatings in a Monitor Pro process display. The upper heating is controlled by thermostat. Its thermostat has been bypassed and its heating is controlled on. The lower heating has a summer/winter switch. It is currently in summer state and has been forced on. On older stations, trace heatings are handled as normal contactors because they only provide the on/off states and controls.

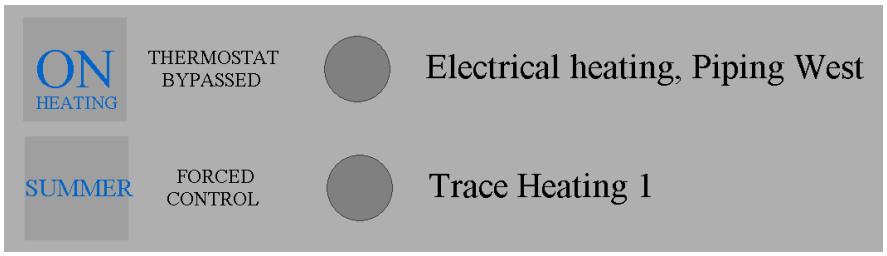

Figure 20. Two trace heating subdrawings in a Monitor Pro process display.

The control dialog of the heating device is set to open when clicking its subdrawing in the process display. Thermostat and season controlled heatings use the same VSO-file. From the control dialog it is possible to check the state of the heating and control it if it's not been locked from the field. The heating type affects the appearance of the control dialog. The control dialog for thermostat-controlled heating type has control buttons for thermostat and continuous heating and the season-controlled heating type has control buttons for summer, winter and forced control.

3.10 Passenger Counting

Passenger counting is an important part of public transport management. Passenger counting offers many benefits for the service provider, including the following [29]:

- Reliable and low-cost data
- Automatically generated reports
- Analyzing of the performance indicators
- Information for traffic scheduling and forecasting
- Comparing passenger numbers with ticket machine transactions
- Monitoring of passenger numbers in the long run

In metro stations, passenger counting is implemented using sensors in every entrance and exit of the station's platform level. The sensors send a pulse to the PLC of the station every time someone walks through them. Pulses from the sensors are then transferred into the MicroSCADA database. The process object type of the sensors must be set to Pulse Counter in order to get correct counting values.

Usually the number of the passengers are wanted to be shown in separate reports and trends. First step is to create measurement reports for the passenger counting. Measurement reports can be found in the Object Navigator. The process objects of the sensors are generated into report objects in the Report Objects folder. After that they are given the correct unit and activated. It is also important to go to Data Objects folder and change the Logging Function of the generated report objects into Pulse Difference.

Next step is to create two new report objects which calculate the sums of the entering and exiting people from the process objects of passenger counting. Final step is to create a report page where the calculated values are shown. The passenger counting report should now be accessible from the Navigation toolbar in the Monitor Pro. The report can also be saved to the hard drive of the MicroSCADA computer with a command procedure which creates a text file of the report into a wanted path.

3.11 Time Programs

Time programs are used in metro applications and the purpose of them is to control certain parts of the metro system at a desired time. Usually the time programs are set to operate in accordance with the opening hours of the stations. Time programs are typically used in doors, PIR sensors and lights of the metro stations.

A command procedure has been created to execute all timed controls in MicroSCADA. First, the command procedure searches all time program process objects with a specific RX (Reserved Text) attribute value. This RX value must be used in every time program related process object. After the search the command procedure checks whether the time program of each object is enabled by checking its XB (Activation Blocking) attribute value and then its execution time from the FI (Free Integer) attribute .

Next phase is to correct the control times based on the day type in the MicroSCADA calendar. Every day type has its own command procedure which changes the FI attribute values of the time transfer process objects. The time program command procedure changes the control execution times based on these FI values. After this the command procedure checks the current time and if the control execution time has already passed, it changes it to the next day. The last step is to execute the timed control within the set time.

The execution times for on and off control for each device are set from their control dialog. It is also possible to enable and disable the object's time programs from it.

4 Engineering Improvement for Metro Applications

Originally one target of this thesis study was to design an engineering improvement to one of the special functions documented above. However, during the documentation of the special functions it turned out that all of them seemed to operate without any problems. All these special functions are currently being used in live systems and seem to work without any issues. Hence, an engineering improvement was proven difficult and unnecessary to accomplish at the moment.

Even though there is currently no need for developing the existing special functions, especially the metro applications have other areas for improvement. Currently one of the main challenges in metro application engineering is the creation of the process displays. Because there is no existing metro device library, most of the metro application functions have had to be hand-made. In addition, new metro stations have a larger number of different devices compared to older stations. It means that these stations require a lot of process displays with a large number of symbols.

In current metro applications, List Based Engineering (LBE) tool has been used to create process displays to MicroSCADA. However, the tool cannot be used in all applications because it requires Microsoft Excel to be installed in the SCADA computer. Therefore, there is a need for creating an alternative solution for the process display engineering. The solution in this thesis focuses on how to efficiently create complete process displays as efficiently as possible without the LBE tool.

The idea was to create similar functionality to the special functions of metro applications as the SA-LIB standard function objects have. SA-LIB stands for The Substation Automation Library and it contains the standard functions for the most generic substation automation objects [30,44]. The subdrawings of the objects created from the standard functions are available from the Object Browser of the Display Editor. They will also have automatically mapped data variables and Tool Launcher configurations which open the correct control dialogs.

4.1 List Based Engineering Tool

The LBE tool is useful when creating process displays based on large databases. It is technically a Microsoft Excel document and it is included in the MicroSCADA installation package. The LBE tool has the following features [30,30]:

- Creation and configuration of process objects
- Creation of graphical objects
- Definition of subdrawing mappings
- Creation of data variables in data sources

The Excel document works as a user interface for the Visual Basic Applications (VBA) macros. The VBA macros perform write operations to the process database based on the information in the Excel sheets. The LBE tool is compatible with Microsoft Excel 2000 or later and it requires a running SYS600 application in the same computer. [30,31.]

4.2 Customizing the Object Browser

The process objects created from the standard functions are available in the Object Browser of the Display Editor. The Object Browser shows the process object hierarchy of the application. From the Object Browser it is possible to choose the correct subdrawing with pre-configured variable mappings for the object and drag it into the process display. [14,169.]

The target was to make the metro application devices available from the Object Browser in a similar way as the standard function objects are. The first step was to find out how to add new process objects into the Object Browser. The objects shown in the Object

Browser are defined in its configuration file. The location of the configuration file is *[drive]\sc\prog\graphicsEngine\etc\ObjNav.ini*. The configuration file identifies the process objects by their Reserved Text (RX) attribute. Every used RX value is given its own symbol definition. The symbol definition consists of the following parts:

- Symbol file path
- Data variable mapping (dynamic and constant)
- Coordinate
- Symbol name

Multiple symbols can be configured to one RX value by adding " | " between the symbol definitions. The coordinate definition is usually left empty, which means that the symbol is created exactly where it is dragged from the Object Browser. Examples of the Bay and Station symbol definitions are shown in Listing 2 below.

```
ABAY_12    =    04 - SA_Indication\Bay.sd:S[10],C[10]::Bay, double indi-
cation | 04 - SA_Indication\Bay Simple.sd:S[10],C[10]::Bay Simple, double
indication

ASTA_12    =    04 - SA_Indication\Station.sd:S[10],C[10]::Station, dou-
ble indication
```

Listing 2. Bay and Station symbol definitions in the Object Browser configuration file.

Next phase was to assign describing RX values to the metro process objects. The given RX values must be equivalent to "RX_Settings" configuration parameters in the configuration file. In this application, "StartIndex" is set to 23 and "Length" is set to 4. It means that the RX values must be four characters long and there must be 22 spaces before the characters in the RX text field. It is not necessary to assign describing RX values for the objects, but it is recommended to avoid unnecessary mistakes. All RX values, subdrawings and data variable mappings used in the customization of the Object Browser configuration were stored in an Excel file to keep track of the progress.

The filters in the configuration file need to be considered when giving the RX values to the process objects in the Object Navigator. In this application, the “IncludeIX” filter has been set to 10. It means that the Object Browser only includes process objects with index 10. However, it is possible to override this filter and give a specified index for a specified RX value by adding the RX and its specified value into the “IncludedSpecifiedIndicesForRxValue” list in the filter configuration.

After the configuration was done, the new RX attribute values were added to the process objects. All objects with configured RX values were now be visible in the Object Browser as shown in Figure 21. The subdrawings of the objects can now be dragged and dropped from the Object Browser into the process displays and they will have correct data variables in them. If other RX values are needed for the process object, it is important to change it back after the objects have been created from the Object Browser. For example, time program related process objects use their own RX value to make the time programs operate correctly.

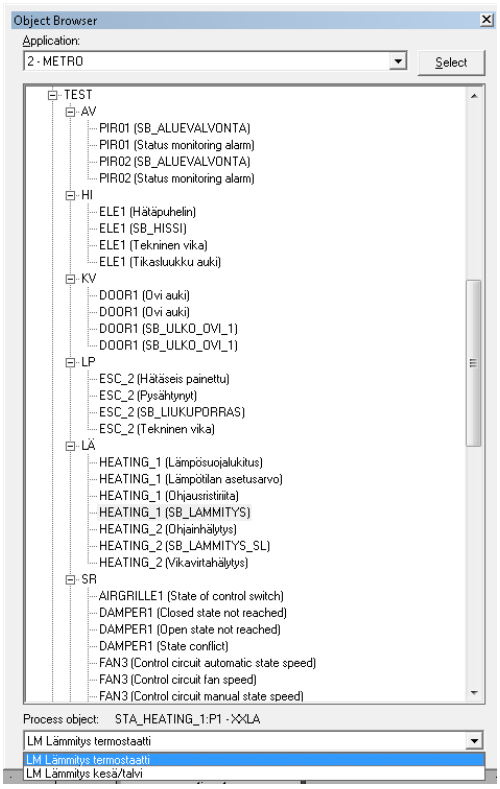

Figure 21. Object Browser with metro application objects.

4.3 Automatic Control Dialog Generation for Subdrawings

The process objects created from the standard functions include the Configuration Data (CD) attribute, which can be found by right-clicking one of the created process objects in the Object Navigator and selecting Group Properties. The configuration data includes the Tool Launcher configuration for the object. It means that every object in the Object Browser created by a standard function will automatically have the correct Tool Launcher settings.

The metro process objects which have been added to be available from the Object Browser do not have this feature, because their CD attributes are empty. It means that the Tool Launcher settings would have to be set manually for each metro application subdrawing that requires a control dialog. This takes a lot of time and is not efficient at all. However, there is a solution to automatically generate correct Tool Launcher settings for metro objects as well.

The Tool Launcher settings can be configured for a single graphical object or for all instances of a graphical symbol. The configuration for all instances is done by right-clicking any instance of a graphical symbol in the Display Editor and selecting Open Subdrawing. After the subdrawing has opened, the Tool Launcher settings for the subdrawing can be configured by right-clicking the drawing area when no graphical objects are selected and selecting Tool Launcher. After the configuration is done, the subdrawing can be saved and closed. [14,47-50.] From now on, the subdrawing will have preconfigured Tool Launcher settings. The Tool Launcher settings were added to every subdrawing used in metro applications that requires a control dialog. The subdrawings with added Tool Launcher settings were also documented into a separate Excel file.

It is important to note that the preconfigured Tool Launcher settings of the subdrawings cannot be modified when using them in process displays. It is only possible to disable the pre-configured Tool Launcher settings and configure new settings only for the selected subdrawing. This means that the “Custom argument” text field in the Tool Launcher Visual SCIL dialog configuration cannot be used because the arguments are usually unique to each object. Usually the custom arguments are used for giving the LN and some additional information of the object. The control dialog needs to know the LN of the object to identify it and to display the right information and control options. Because the custom arguments cannot be used, another solution must be devised to find the correct LN for each object.

Examining the SCIL methods of the VSO-files for metro application objects revealed that no custom arguments were used in the control dialogs for modern metro objects. However, when tested with preconfigured Tool Launcher settings these control dialogs gave an error code and did not open. A closer examination of the SCIL methods of the control dialogs revealed that the SCIL program retrieves the LN of the object from the Object Name attribute of the subdrawing. It means that every subdrawing must have the same Object Name as the LN of the object it presents. After renaming the subdrawings, the control dialogs opened without errors and functioned as they should.

4.4 Testing of the Engineering Improvement

After the changes had been made, it had to be ensured that they functioned as desired. The testing of these improvements was done by creating new test objects into the MicroSCADA database and adding them into a test process display. The testing was done in a virtual computer located in the ABB Power Grids office in Helsinki. The MicroSCADA version in the virtual computer was SYS600 9.4 FP2 HF1. There was no need to test these changes in a live MicroSCADA system because they are designed to function in every MicroSCADA application. The type of the improvements also means that no live system is required to test their functionality.

Each subdrawing for every object was dragged one by one from the Object Browser to the test process display. Finnish versions of the subdrawings were used because English versions were not available for every metro application object at the time. It was then checked that the state indications of the subdrawings appeared correctly in the process display. The Tool Launcher configurations were also checked from the subdrawings that require a control dialog. Finally, the Object Name attributes were changed to match the LN of the object it presents. A few mistakes from the Object Browser configuration file

were found during the testing. These mistakes were fixed after which all the made changes started to function as planned. The process display with the test objects is shown in Figure 22.

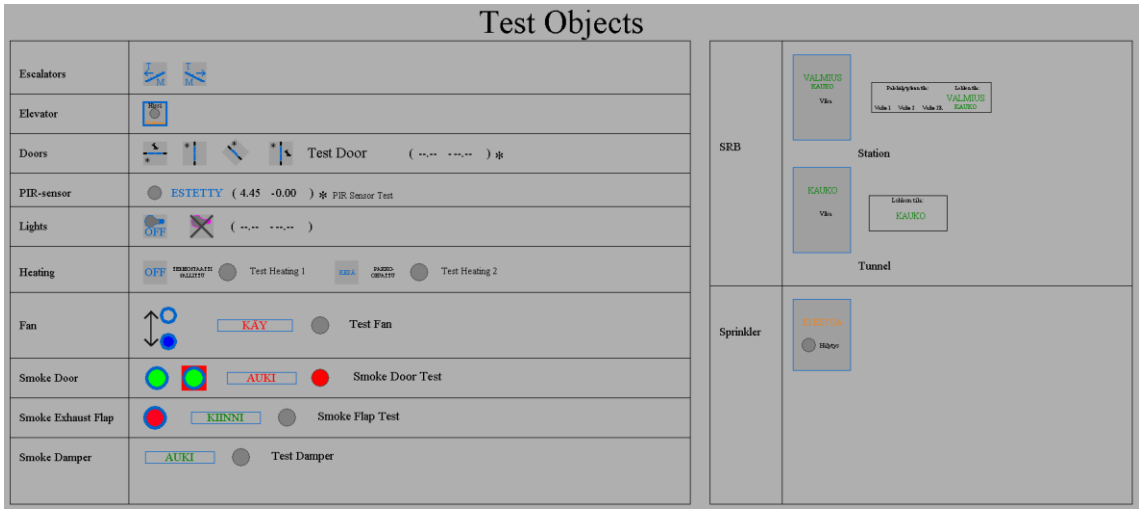

Figure 22. A Monitor Pro process display with the metro application test objects.

5 Summary

The result of this thesis study is a comprehensive documentation of the special functions and engineering phases needed in MicroSCADA metro and railway applications. Also, an alternative solution for process display creation was implemented. Most of the documented special functions concerned the metro applications, because the metro applications include a lot of building automation which MicroSCADA is not designed for. Most of the features in the current railway applications are already available in MicroSCADA, which has resulted in lower amount of special functions needed to be created.

The target of the engineering improvement was to add similar functionality to the special functions of metro applications as the SA-LIB standard functions have. Customizing the Object Browser and adding the Tool Launcher configurations to the subdrawings made the special functions easier to use. The changes also speed up creation of the process displays compared to fully manual creation. In addition to the metro applications, these improvements can be used every MicroSCADA application that requires special subdrawings and control dialogs, which are not available in MicroSCADA libraries.

Even though these improvements significantly speed up the process display engineering compared to manual creation, the LBE tool is still a better and faster solution for the creation of the process displays. If the application has a large database, the number of objects in the Object Browser would be massive. It would be difficult to search the correct objects from the Object Browser and the possibility of errors increases at the same time. With the LBE tool it is possible to create process displays that include all wanted subdrawings with correct variable mappings and Tool Launcher configurations by a couple of mouse clicks.

The improvements made in this thesis study are still a good alternative solution for the LBE tool when Microsoft Excel is not available in the SCADA computer. Perhaps the biggest benefit of these improvements is obtained when updating the process displays of an existing application. It will be easy to add new device symbols to existing process displays when they are available straight from the Object Browser. Time will tell whether these improvements will be used in existing and future MicroSCADA applications.

The number of special functions and engineering phases related to them could be significantly reduced if MicroSCADA would support more applications areas. Also, there would be no need for the engineering improvements created in this thesis study. MicroSCADA has been on the market for decades. All this time it has been developed

mainly for substation and process industry applications. However, it is proven to be a good solution for other applications as well. It means that MicroSCADA is able to handle all kinds of signals and devices. This has not yet been fully utilized since the MicroSCADA object library includes symbols mainly for substation and process industry applications. In addition, the number of available standard functions could be much larger. The content of this thesis study demonstrates well the amount of engineering work required to make MicroSCADA suitable for an application for which it was not originally designed for. Considering future MicroSCADA applications, the object library should be expanded to cover more application areas. Better support for different application areas would increase customer interest in MicroSCADA while reducing the engineering work of those application areas.

References

1 ABB Power Grids Finland Oy. 2019. Uusi energia-alan edelläkävijäyhtiö ABB Power Grids Finland Oy käynnisti toimintansa Suomessa [online]. URL: https://new.abb.com/news/fi/detail/41237/uusi-energia-alan-edellakavijayhtio-abb-power-grids-finland-oy-kaynnisti-toimintansa-suomessa. Accessed 9 December 2019.

2 Mutnukrishnan Vidya. 2019. SCADA System: what is it? (Supervisory Control and Data Acquisition) [online]. Electrical4U. URL: https://www.electrical4u.com/scada-system/. Accessed 27 November 2019.

3 UpKeep. 2019. What is a SCADA system [online]? URL: https://www.onupkeep.com/learning/maintenance-tools/scada-system. Accessed 10 December 2019.

4 Anderson Mondi. 2019. What is SCADA? [online]. RealPars B.V. URL: https://realpars.com/scada/. Accessed 10 December 2019.

5 ABB Oy, Substation Automation Products. 2004. MicroSCADA Pro SYS 600 *9.0 System Overview – Technical Description. Manual.

6 Österbacka CG, ABB Oy Substation Automation. 2002. The Birth of MicroSCADA.

7 ABB Oy, Grid Automation Products. 2016. MicroSCADA Pro SYS600 9.4 System Configuration. Manual.

8 ABB Oy, Grid Automation Products. 2016. MicroSCADA Pro SYS600 9.4 Application Objects. Manual.

9 ABB Oy, Grid Automation Products. 2018. All-inclusive power automation solution. MicroSCADA Pro SYS600C. Brochure.

10 ABB Oy. MicroSCADA Pro DMS600 [online]. URL: https://new.abb.com/network-management/network-management/microscada-pro/dms600. Accessed 8 January 2020.

11 ABB Oy. 2017. MicroSCADA Pro SYS600 9.4 Application Design. Manual.

12 ABB Oy, Grid Automation Products. 2016. MicroSCADA Pro SYS600 9.4 System Objects. Manual

13 ABB Oy, Substation Automation Products. 2014. MicroSCADA Pro SYS600 9.4 Visual SCIL Application Design. Manual.

14 ABB Oy, Grid Automation Products. 2016. MicroSCADA Pro SYS600 9.4 Process Display Design. Manual.

15 ABB Oy. 2016. MicroSCADA Pro SYS600 9.4 Programming Language SCIL. Manual.

16 Wienand Ian. 2016. Computer Science from the Bottom Up [online]. URL: https://www.bottomupcs.com/index.xhtml. Accessed 2 December 2019.

17 Santaharju Pentti. 1973. Raitiotievaihteiden sähkölämmitys Helsingissä [online]. SRS Ry. URL: https://www.raitio.org/vanhasivusto/historia/vaihteet/sulatus.htm. Accessed 31 December 2019.

18 Ratahallintokeskus. 2006. Vaihteenlämmityksen tekniset määreet.

19 Li Ivy. 2019. PIR Sensor Introduction and How PIR Motion Sensor works with Arduino and Raspberry Pi [online]. URL: https://www.seeedstudio.com/blog/2019/08/03/pir-sensor-introduction-and-how-pir-motion-sensor-works-with-arduino-and-raspberry-pi/. Accessed 3 December 2019.

20 Airamaa Minna. 2018. Länsimetro lyö oman ennätyksensä – Espooseen Suomen pisimmät liukuportaat [online]. Länsiväylä. URL: https://www.lansivayla.fi/artikkeli/706038-lansimetro-lyo-oman-ennatyksensa-espooseen-suomen-pisimmat-liukuportaat. Accessed 29 November 2019.

21 Laine Lauri. 2019. LED-valojen ohjaus – DALI [online]. URL: https://www.winled.fi/blogi/artikkeli/Led-valojen-ohjaus-—-DALI. Accessed 5 December 2019.

22 Hengitysliitto. Ilmanvaihto [online]. URL: https://www.hengitysliitto.fi/fi/sisailma/ilmanvaihto. Accessed 10 December 2019.

23 Pohjanmaan pelastuslaitos. Savunpoisto [online]. URL: https://www.pohjanmaanpelastuslaitos.fi/palvelut/rakenteellinen-paloturvallisuus/savunpoisto. Accessed 10 December 2019.

24 Wikipedia. Smoke dampers [online]. URL: https://en.wikipedia.org/wiki/Smoke_dampers. Accessed 11 December 2019.

25 Rossi et al. 2012. Länsimetron jatke, hankesuunnitelma Matinkylä-Kivenlahti [online]. URL: https://www.lansimetro.fi/wp/wp-content/uploads/2017/03/1_matinkyla-kivenlahti_metro-hankesuunnittelu_web.pdf. Accessed 28 January 2020.

26 Champion Door Oy. Palo-ovet ja savuverhot [online]. URL: https://www.championdoor.com/fi/tuotteet/palo-ovet-ja-savuverhot. Accessed 11 December 2019.

27 Kirn Lucas. 2019. How Does a Fire Sprinkler System Work [online]? Engineered Corrosion Solutions. URL: https://www.ecscorrosion.com/blog/how-does-a-fire-sprinkler-system-work. Accessed 13 December 2019.

28 Supermec Pte. Ltd. 2019. What is trace heating [online]? URL: https://www.supermec.com/it/blog/2019/01/28/what-is-trace-heating. Accessed 16 December 2019.

29 Retail Sensing Ltd. 2019. Automatic Passenger Counting [online]. URL: https://www.retailsensing.com/automated-passenger-counting.html. Accessed 9 December 2019.

30 ABB Oy, Substation Automation Products. 2008. MicroSCADA Pro SYS600 9.2 Application Design. Manual.