VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Matti Jaatinen

SIMULATION SYSTEM COOPERATION WITH SCADA/DMS

Technology and Communication
2016

VAASAN AMMATTIKORKEAKOULU
Sähkötekniikan koulutusohjelma

TIIVISTELMÄ

Tekijä	Matti Jaatinen
Opinnäytetyön nimi	Simulointiohjelman liitettävyys SCADA-järjestelmän kanssa
Vuosi	2016
Kieli	englanti
Sivumäärä	31
Ohjaaja	Jari Koski

Tänä päivänä SCADA-järjestelmän vaatimuksiin kuuluu tyypillisesti simulointitoiminnot operaattorin harjoittelua varten. MicroSCADA ja sähköverkkojen analysointiohjelma yhdessä pystyvät vastaamaan tähän. Tämän työn tarkoituksena oli tutkia MicroSCADAn ja sähköverkkojen analysointiohjelman liitettävyyttä. Neplan valikoitui sähköverkkojen analysointiohjelmaksi.

Projektiasiakkaiden toimesta on herännyt kiinnostusta harjoittelumoodiin MicroSCADA-järjestelmässä. Tästä syystä on todettu tarve tutkia mitä vaihtoehtoja ja vaatimuksia on tarvittavan rajapinnan rakentamiseen. Tässä opinnäytetyössä testattiin erilaisten rajapintojen toimivuutta myös käytännössä.

Rajapinta oli mahdollista rakentaa MicroSCADAn ja Neplanin välille, joko rajapintatiedostomenetelmällä tai SQL-tietokantamenetelmällä. Ongelmaksi muodostui kuitenkin rajapinnan toimiminen automaattisesti. Tulevaisuudessa voitaisiin tutkia mm. Neplan 360-ohjelman tai Etap-ohjelman liitettävyyttä Micro-SCADA-järjestelmään.

Avainsanat MicroSCADA, Neplan, ODBC, NPL, Käyttöliittymä, SQL tietokanta

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Sähkötekniikan koulutusohjelma

ABSTRACT

Author	Matti Jaatinen
Title	Simulation System Cooperation with SCADA/DMS
Year	2016
Language	English
Pages	31
Name of Supervisor	Jari Koski

Today the requirements for SCADA/DMS typically include simulation functions for operator training. Together MicroSCADA Pro and network analysis software can respond for this requirement. The purpose of the thesis was to investigate connectivity between MicroSCADA and network analysis software. Neplan was selected to work as a NA software.

The project customers' interest has increased towards training mode in MicroSCADA system. For this reason there is a need to investigate different possibilities and requirements for building an interface. In this thesis different interface possibilities were also tested in practice.

It was possible to establish an interface between MicroSCADA and Neplan, either with the dynamic interface file method or with the SQL database method. The problem was that the interface did not work automatically. In the future Neplan 360 or Etap could be taken under the investigation.

Keywords MicroSCADA, Neplan, ODBC, NPL, Interface, SQL database

TABLE OF CONTENTS

ABBREVIATIONS

HMI	Human-Machine interface
LF	Load flow
.SCE	File format for data exchange
NPL	Neplan Programming Library
API	Application Program Interface
SQL	Structured Query Language
ANSI	American National Standard Institute
ISO	International Standard Organisation
ODBC	Open Database Connectivity
IED	Intelligent Electronic Device
NA	Network analysis
SCIL	Supervisory Control Implementation Language
OPC	Open Platform Communications

LIST OF FIGURES

1 INTRODUCTION

This thesis was made for ABB Grid Automation to investigate an available interface between SCADA and network analysis software. Today SCADA system requirements typically include simulation functions for operator training and simulation purposes. MicroSCADA does not have these functions and on the other hand NA analysis software does not have all MicroSCADA functions. The interface between MicroSCADA and simulation software could make this requirement possible.

The SCADA application and the simulation software HMIs should be as similar as possible. It would be very important to get the smart calculation model from the simulation software integrated with MicroSCADA. The simulation software Neplan was used in this thesis.

The first part of the thesis introduces the basic knowledge about MicroSCADA and Neplan. After that we concentrate on an interface between Neplan and the external SCADA system where the requirements for the interface are introduced. The next chapters introduce the interfaces that are supported by Neplan. They are the dynamic interface file and the SQL database. ODBC and NPL functionality with these interfaces are presented after that. The last part of this thesis presents results how these interfaces work.

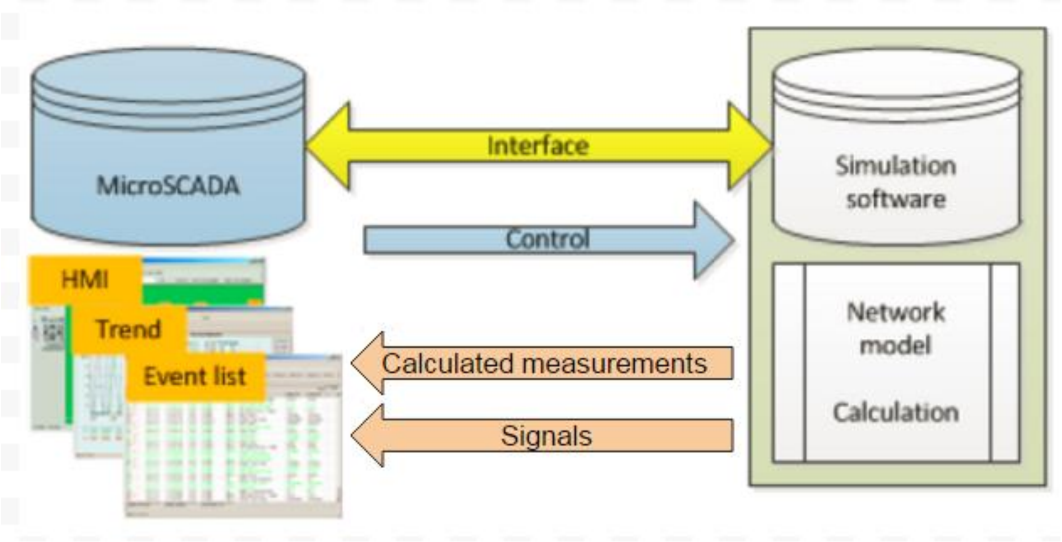


Figure 1. Simulation system main components.

2 ABB

ABB is a worldwide leader in power and automation technologies. The headquarters located in Zurich, Switzerland. The company employs 135 000 people in 100 countries. /1/

2.1 ABB in Finland

Strömberg was established in 1889 by Gottfrid Strömberg. Electrical machines planned by Strömberg won many prices in international exhibitions. Vaasa factories were established in 1944. In 1987 whole part of Strömberg sold to a Swedish company Asea after which Asea soon merged with a Swiss company Brown Boveri in 1988. As a result, ABB was established. In Finland ABB Oy employs approximately 5100 people. The factories are located in Helsinki, Vaasa and Porvoo. ABB Oy uses a lot of resources on product development. /2-5/

2.2 ABB Grid Automation

In Finland ABB Grid Automation business unit develops and delivers Supervision Control and Data Acquisition systems (SCADA). SCADA supervises and controls transmission and distribution networks. The Grid Automation business unit also offers training services, around-the-clock technical support and maintenance. Grid Automation employs approximately 100 people in Finland and it belongs under the Power Systems division. /14/

3 REQUIREMENTS FOR COOPERATION BETWEEN MICROSCADA AND NETWORK ANALYSIS SOFTWARE

3.1 MicroSCADA pro

MicroSCADA Pro is designed to monitor and control of primary and secondary equipment in transmission and distribution substations in a real time. MicroSCADA Pro is easy to interact with protection and control IEDs. With MicroSCADA circuit breakers, disconnectors and tap changers can be controlled. Additional control functions is also possible to create. MicroSCADA Pro supports the IEC 61850 standard for substation automation. IEDs, tools and systems which are supported with IEC 61850 can operate together. That simplifies system engineering. /17/

3.2 Network analysis system

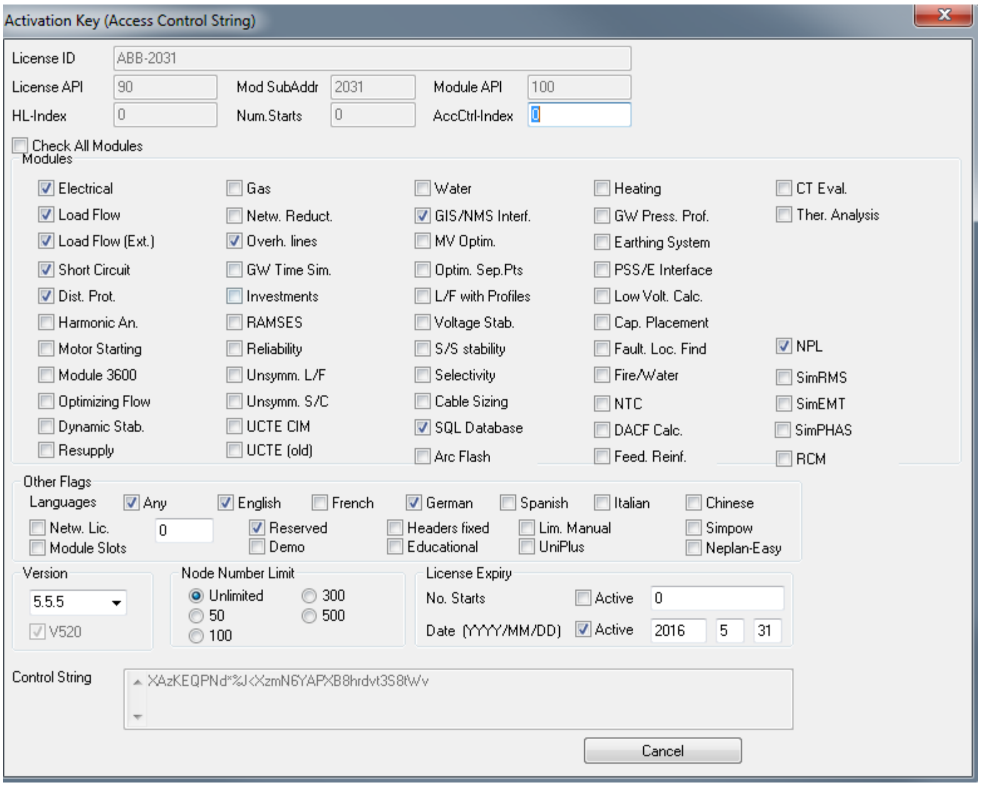
Based on customers' specification there are lot of requirements for network analysis software. The network analysis software should work as an online predictive simulator based on response to operator actions and events via the use of real-time and archived data from analysis modules, such as load flow, short-circuit and motor starting. The simulator functions shall include the following features and capabilities:

1. Collection, upload, display and archiving of the time stamped data
2. Simulation should be possible at any time under various system configurations.
3. Training mode for operators
4. Simulate the operation of system equipment such as relays, fuses, circuit breakers etc.
5. Network colouring for faults and disturbances.
6. Playback of previously recorded events.
7. Real-Time Operation of System Components
8. Full spectrum network analyse modules

These requirements are just one example. Some of them are possible to be put into practice. It is hard to respond to all the above mentioned requirements. The training mode for operators is the most important function to get working. Today SCADA system requirements typically include simulation functions for operator training and simulation purposes. For that reason cooperation between network analysis program and SCADA is needed. Neplan software responds quite well for these requirements. /15/

4 NEPLAN ELECTRICITY

Neplan Electricity is a software tool to analyse, plan, optimize and simulate electrical networks. The Neplan software is based on a modular concept which covers all electrical aspects in transmission, distribution, generation and industrial networks. Neplan is mainly used with a graphical interface. Neplan has various modules to choose from for example Load Flow calculation, Short Circuit calculation, SQL Database Driver (ODBC), SQL Database Converter, interface to SCADA and GIS systems. The chosen modules have an individualized licence to fit the needs of the company. Neplan can be used as a desktop application, and as a web service Neplan 360. Neplan 360 is a browser based power system analysis tool. Neplan 360 works in cloud or intranet. The desktop application was used in this thesis. /6/



Activation Key (Access Control String)

License ID ABB-2031
License API 90 Mod SubAddr 2031 Module API 100
HL-Index 0 Num.Starts 0 AccCtrl-Index 0

□ Check All Modules

Modules

☑ Electrical	□ Gas	□ Water	□ Heating	□ CT Eval.
☑ Load Flow	□ Netw. Reduct.	☑ GIS/NMS Interf.	□ GW Press. Prof.	□ Ther. Analysis
☑ Load Flow (Ext.)	☑ Overh. lines	□ MV Optim.	□ Earthing System	
☑ Short Circuit	□ GW Time Sim.	□ Optim. Sep.Pts	□ PSS/E Interface	
☑ Dist. Prot.	□ Investments	□ L/F with Profiles	□ Low Volt. Calc.	
□ Harmonic An.	□ RAMSES	□ Voltage Stab.	□ Cap. Placement	
□ Motor Starting	□ Reliability	□ S/S stability	□ Fault. Loc. Find	☑ NPL
□ Module 3600	□ Unsymm. L/F	□ Selectivity	□ Fire/Water	□ SimRMS
□ Optimizing Flow	□ Unsymm. S/C	□ Cable Sizing	□ NTC	□ SimEMT
□ Dynamic Stab.	□ UCTE CIM	☑ SQL Database	□ DACF Calc.	□ SimPHAS
□ Resupply	□ UCTE (old)	□ Arc Flash	□ Feed. Reinf.	□ RCM

Other Flags
Languages ☑ Any ☑ English □ French ☑ German □ Spanish □ Italian □ Chinese
□ Netw. Lic. 0 ☑ Reserved □ Headers fixed □ Lim. Manual □ Simpow
□ Module Slots □ Demo □ Educational □ UniPlus □ Neplan-Easy

Version
5.5.5
☑ V520

Node Number Limit
● Unlimited ○ 300
○ 50 ○ 500
○ 100

License Expiry
No. Starts □ Active 0
Date (YYYY/MM/DD) ☑ Active 2016 5 31

Control String XAzKEQPNd*%J<XzmN6YAPXB8hrdvt3S8tWv

Figure 2. Modules used in this thesis. /18/

4.1 Network Planning with Neplan

Planning works with a graphical interface where networks can be drawn in a single line diagram form. Electrical data in HMI is built on component basis so experience with electrical networks is needed. The Neplan project used in this thesis was taken from the Industrial and Power Plant Electrical Systems course. For that reason there is no needed to build an electrical network. Drawing a single line diagram for the Neplan was quite easy, just drag and drop components on the drawing area. Difficulties come from what parameters are inside electrical components. Some of parameters are hard to understand, what effect they have on the electrical network, still if the user has some experience about electrical networks when using Neplan first time. There are not explanations for every parameter. That is why effects of parameters need testing quite a lot. For that reason the Neplan project was used which is ready for simulation. The simulation is quite easy when the network is ready and defined in a correct way.

4.2 Load Flow Analysis

The default calculation method in a load flow is an extended Newton-Raphson method. Other available options are the Newton-Raphson method, current iteration and detection for voltage drop. LF results can be exported to an SQL database or to an ASCII file that can be opened with MS-Excel. It is also possible to get the resulted text file exported automatically after every LF calculation. /7/

4.3 Short-circuit Analysis

In the short-circuit analysis there are possibilities to use different computation methods, for example, IEC 909/VDE0102, ANSI/IEEE, IEC 60909, superposition method. The fault type can be chosen between 3-phase fault, 1-phase to ground fault, 2-phase fault, 2-phase to ground fault, special fault and fault between all existing phases. /7/

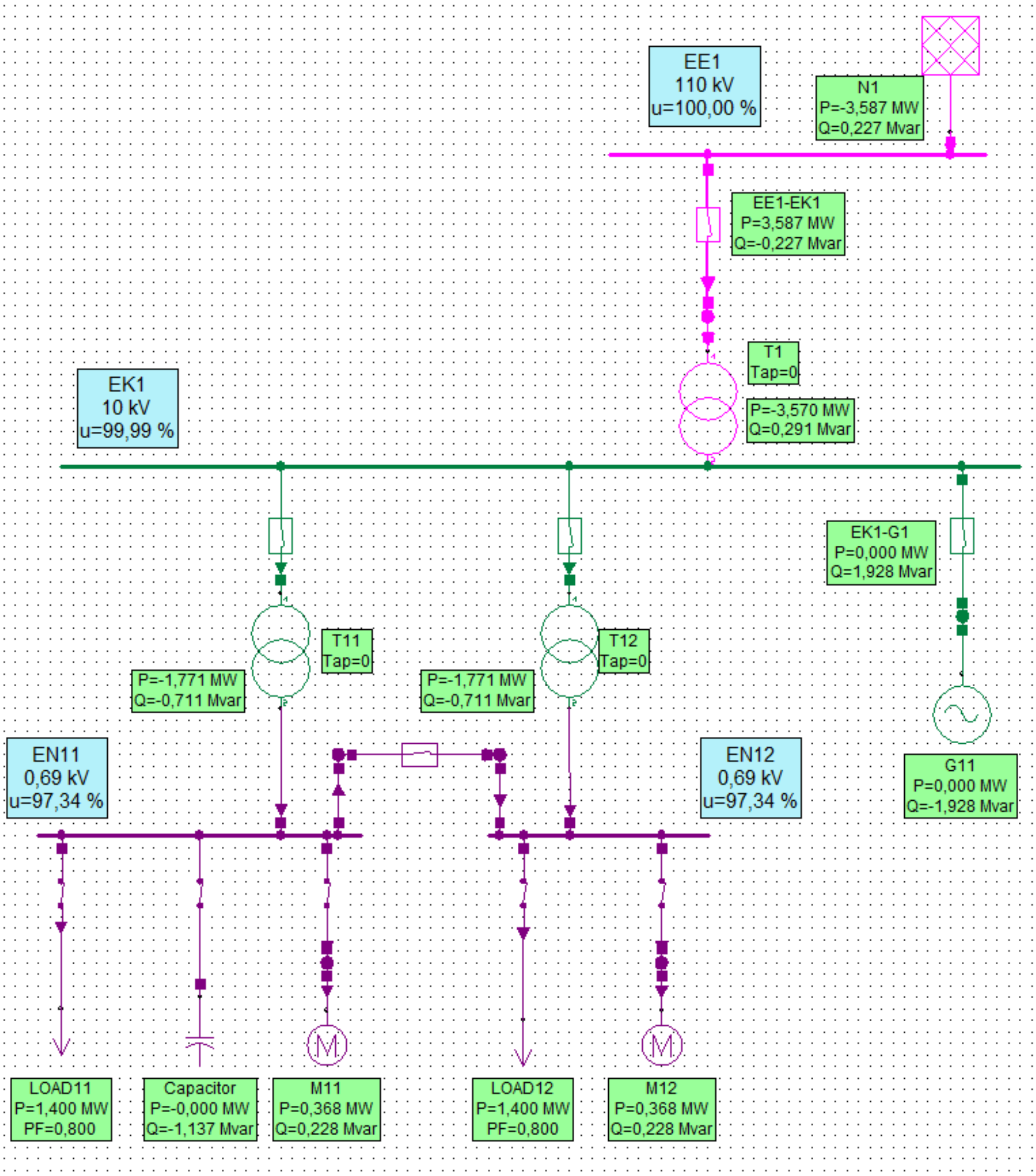


Figure 3. Network in Neplan.

5 INTERFACE BETWEEN THE NEPLAN AND EXTERNAL MICROSCADA SYSTEM

MicroSCADA and Neplan use their own databases for storing the process data and the graphical data of a network. It is obvious that the data is presented in different ways between these applications. In theory there are two possibilities to avoid this problem:

- To create one common database for applications.
- To leave applications work with their own databases and get an interface that works as a data transfer tool between these databases.

There are no international standards of how the network data should be stored in the database. For example, if calculation standards will change in the future, a common database definition will be difficult. For that reason it is a good thing that databases are kept independent between different applications. Therefore, there must be an interface to transfer and update the network data. /8/

5.1 Requirements for Interface

The interface must be able to transfer the state of the switches and taps of on-load tap changer or normal tapped transformers. Measurements are also very important to transfer further, such as power of loads, reactive powers and currents. It is desirable that the interface works as fast as possible. The interface should be kept as simple as possible for functionality reasons. A simpler interface is usually much easier to keep reliable. /8/

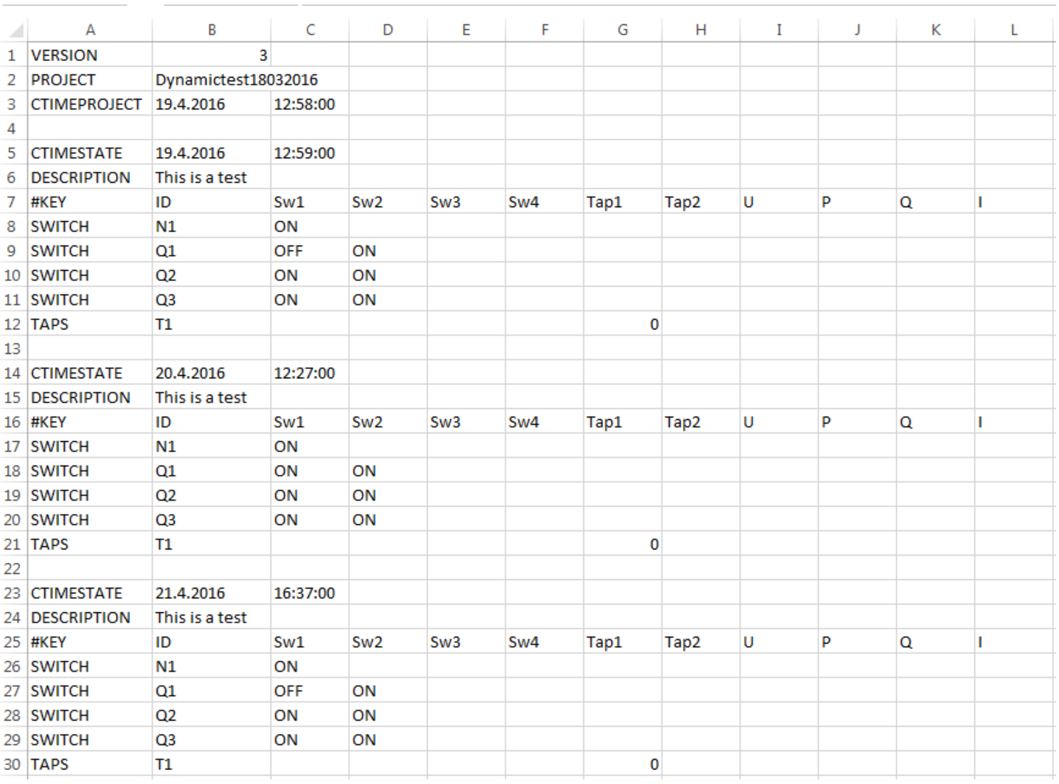
5.2 Available Interfaces between MicroSCADA Pro and Neplan

There are two possibilities for an interface. They are the dynamic interface file and SQL interface. MicroSCADA Pro needs to write control data to the dynamic interface file when a new control command is given on the SCADA HMI. The LF result file must be able to read. The SCADA HMI must show the results in the LF file.

These functionalities are possible to do with the SCIL code in MicroSCADA. SCIL stands for Supervisory Control Implementation Language. SCIL is designed for the application engineering of the supervisory control system SYS600. The other possibility is to take a connection with the SCIL code to the SQL database and give the control commands through it. All measurement results are also located in the SQL database. The circulation in MicroSCADA could be for instance when the operator changes the state of the network, for example changes the state of the circuit breaker. After that MicroSCADA writes a new control value data to the dynamic interface file or to the SQL database. There are functions in SCIL to read and write text files.

5.3 Interface file

The position indications of the switches and transformers will be transferred through the dynamic interface file which is delimited by tabs. This file will be imported into Neplan when the state of the network is changed in MicroSCADA. Neplan allows only unique names for electrical components which has to be taken into account in the ID name in the dynamic interface file. The interface file is in .SCE format that is the data exchange format. In this thesis the file was created with MS-Excel. (Figure 4). Neplan cannot import the file automatically so NPL (Neplan programming library) is needed for that.



	A	B	C	D	E	F	G	H	I	J	K	L
1	VERSION	3										
2	PROJECT	Dynamictest18032016										
3	CTIMEPROJECT	19.4.2016	12:58:00									
4												
5	CTIMESTATE	19.4.2016	12:59:00									
6	DESCRIPTION	This is a test										
7	#KEY	ID	Sw1	Sw2	Sw3	Sw4	Tap1	Tap2	U	P	Q	I
8	SWITCH	N1	ON									
9	SWITCH	Q1	OFF	ON								
10	SWITCH	Q2	ON	ON								
11	SWITCH	Q3	ON	ON								
12	TAPS	T1					0					
13												
14	CTIMESTATE	20.4.2016	12:27:00									
15	DESCRIPTION	This is a test										
16	#KEY	ID	Sw1	Sw2	Sw3	Sw4	Tap1	Tap2	U	P	Q	I
17	SWITCH	N1	ON									
18	SWITCH	Q1	ON	ON								
19	SWITCH	Q2	ON	ON								
20	SWITCH	Q3	ON	ON								
21	TAPS	T1					0					
22												
23	CTIMESTATE	21.4.2016	16:37:00									
24	DESCRIPTION	This is a test										
25	#KEY	ID	Sw1	Sw2	Sw3	Sw4	Tap1	Tap2	U	P	Q	I
26	SWITCH	N1	ON									
27	SWITCH	Q1	OFF	ON								
28	SWITCH	Q2	ON	ON								
29	SWITCH	Q3	ON	ON								
30	TAPS	T1					0					

Figure 4. Example for dynamic interface file.

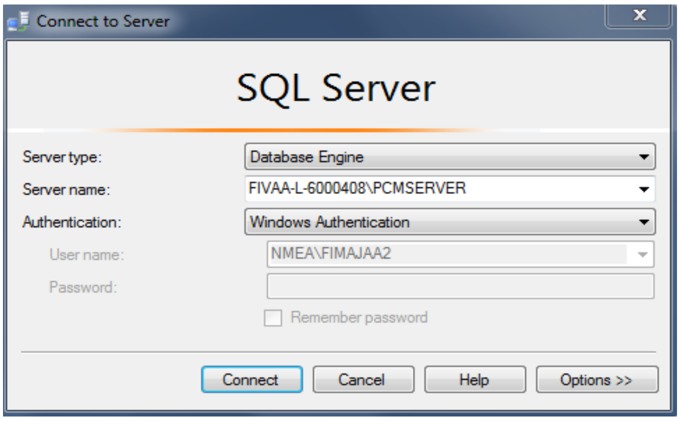
6 SQL INTERFACE

6.1 SQL – Structured Query Language

SQL stands for Structured Query Language which is used to communicate with a relational databases. ANSI and ISO have accepted SQL as the standard language for relational databases. In the previous paragraph it was mentioned how the dynamic interface file will work as an interface. The other possibility for creating an interface is to use SQL. There is an option to export data from Neplan to SQL database. SQL database has to be created. /11/

6.2 Creating SQL Database and ODBC Driver

In this thesis Microsoft SQL server 2016 Management studio was used for creating SQL database. With the SQL server it is easy to create a new database. When the SQL server is opened a “connect to- server“ window appears where a server type and authentication need to be defined. The database engine needs to be used as a server type and the Windows authentication is used as an authentication. The SQL server authentication should be used when connecting to external networks. The server name was there already which is computer name\PCMSERVER. After pressing Connect, a new database must be created by right-clicking Database folder so right-click it and clicking New database. It is not still possible to transfer a data between Neplan and SQL database.



Connect to Server

SQL Server

Server type: Database Engine

Server name: FIVAA-L-6000408\PCMSERVER

Authentication: Windows Authentication

User name: NMEA\FIMAJAA2

Password:

□ Remember password

Figure 5. Connect to server window in SQL server 2016.

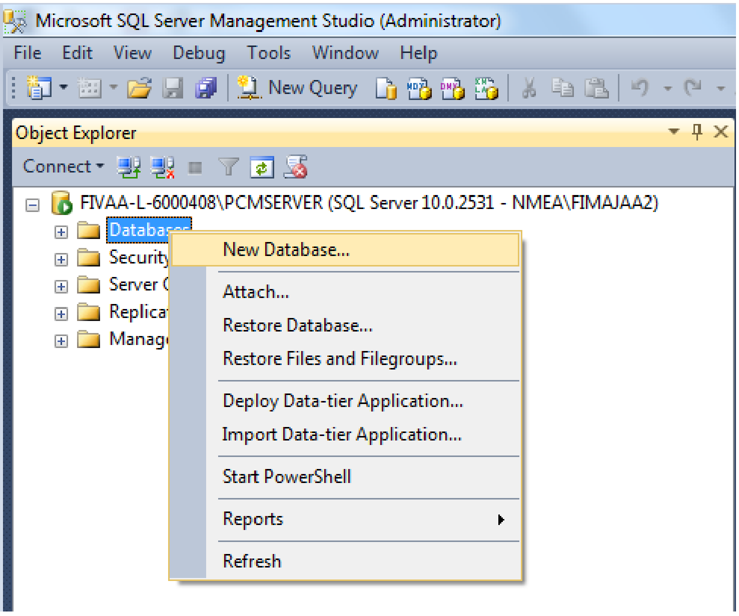
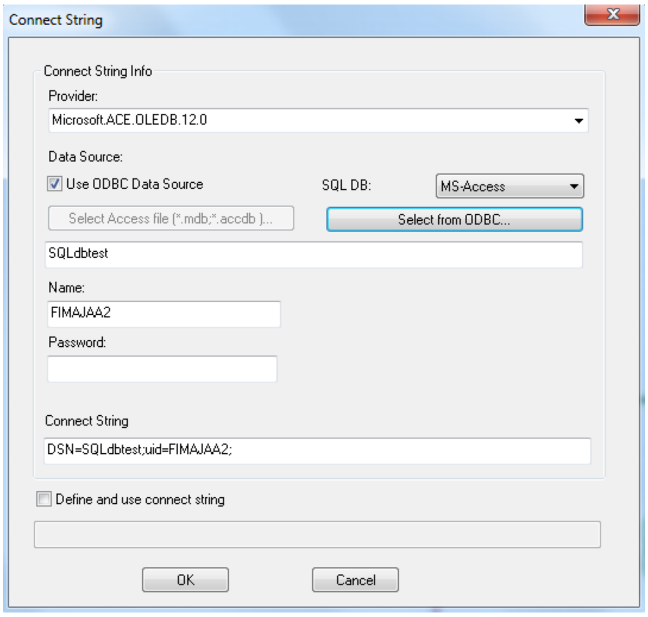

Microsoft SQL Server Management Studio (Administrator)
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect
FIVAA-L-6000408\PCMSERVER (SQL Server 10.0.2531 - NMEA\FIMAJAA2)
Databases
Security
Server
Replica
Manag
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh

Figure 6. New Database.

ODBC needs to be used which works an interface to data transferring between Neplan and the SQL database. The ODBC interface can access data between different database management systems. ODBC is designed specifically for relational data store so the ODBC driver must be installed. Installation for the driver is located in Control Panel\All Control Panel Items\Administrative Tools\Data sources (ODBC). When installing the driver the driver for SQL server needs to be chosen because that is the program that was used for creating the SQL database. When the driver is created, the Neplan is ready to communicate with the SQL database. That happens by clicking “FILE -> EXPORT -> SQL DATABASE”. The ODBC driver needs to be chosen that was already done, so Neplan defines the connect string in the right way automatically. The last step is the SQL export window Delete database prior to export needs to be defined and save node ID’s or names for the topology. It is also possible to choose what data needs to be exported.



Connect String Info

Provider: Microsoft.ACE.OLEDB.12.0

Data Source:

☑ Use ODBC Data Source

SQL DB: MS-Access

SQLdbtest

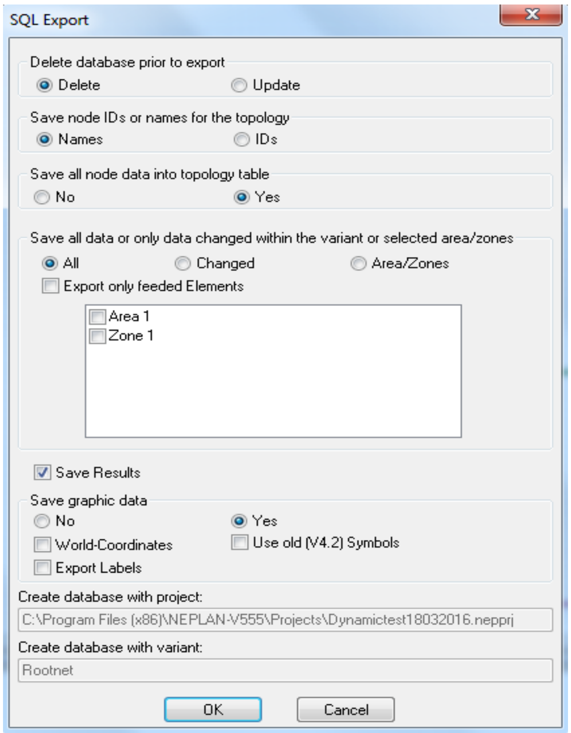
Name: FIMAJAA2

Password:

Connect String: DSN=SQLdbtest;uid=FIMAJAA2;

□ Define and use connect string

Figure 7. SQL export with Neplan.



Delete database prior to export
● Delete ○ Update

Save node IDs or names for the topology
● Names ○ IDs

Save all node data into topology table
○ No ● Yes

Save all data or only data changed within the variant or selected area/zones
● All ○ Changed ○ Area/Zones

□ Export only feeded Elements

□ Area 1
□ Zone 1

☑ Save Results

Save graphic data
○ No ● Yes
□ World-Coordinates □ Use old (V4.2) Symbols
□ Export Labels

Create database with project:
C:\Program Files (x86)\NEPLAN-V555\Projects\Dynamictest18032016.nepprj

Create database with variant:
Rootnet

Figure 8. SQL export in Neplan.

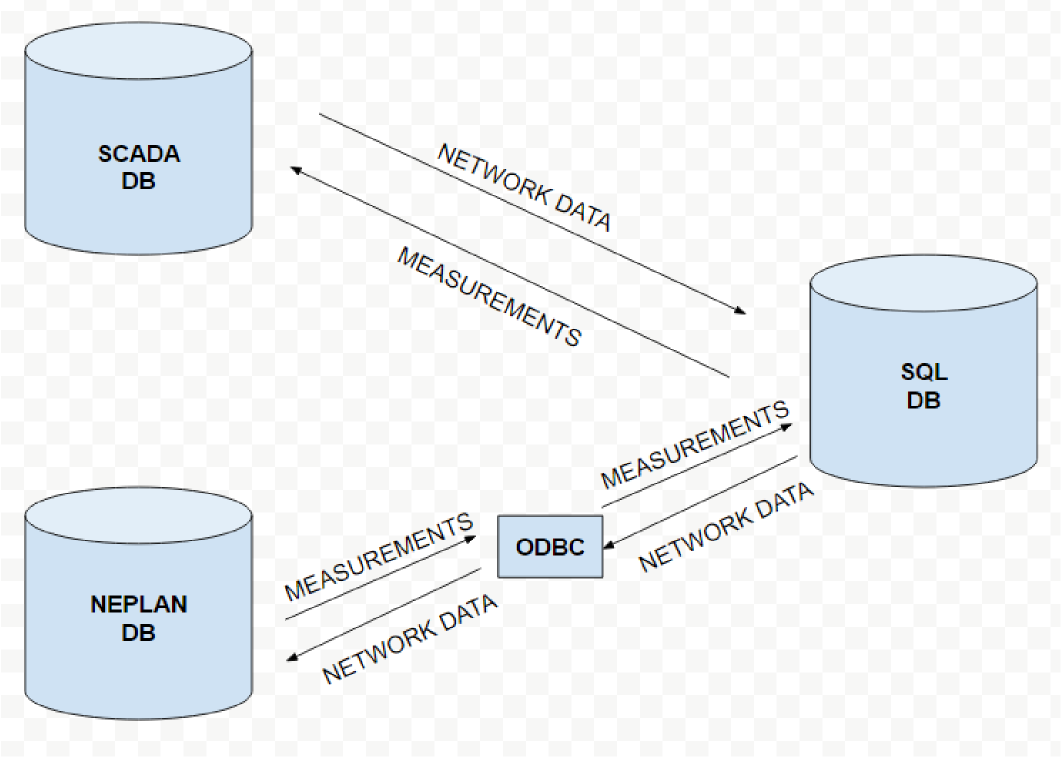
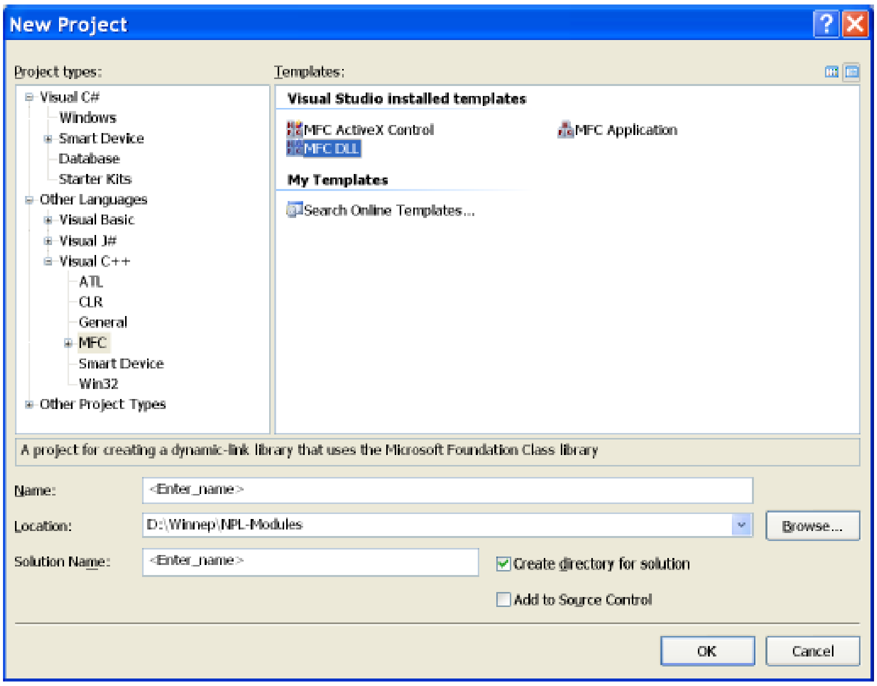


Figure 9. Communication chart.

7 NPL – NEPLAN PROGRAMMING LIBRARY

The NPL – Neplan Programming Library is a C/C++ API library. The library consist of functions for accessing Neplan data and calculation algorithms through a C/C++ user written program. With these functions it is possible to run any calculation analysis functions, such as load flow analysis, transient stability analysis, and short circuit analysis. It also allows to change the length of the line or to modify the short circuit voltage of a transformer. NPL can manipulate Neplan projects through C/C++ program. The C/C++ programmer can define an own “batch program”. The programmer needs to include a NeplanProgrammingLibrary.h header file into the C++ project and link it with NPL NeplanProgrammingLibrary.lib. In Microsoft Visual studio new MFC DLL project must be created. /10/



A project for creating a dynamic-link library that uses the Microsoft Foundation Class library

Name: <Enter_name>

Location: D:\Winnep\NPL-Modules

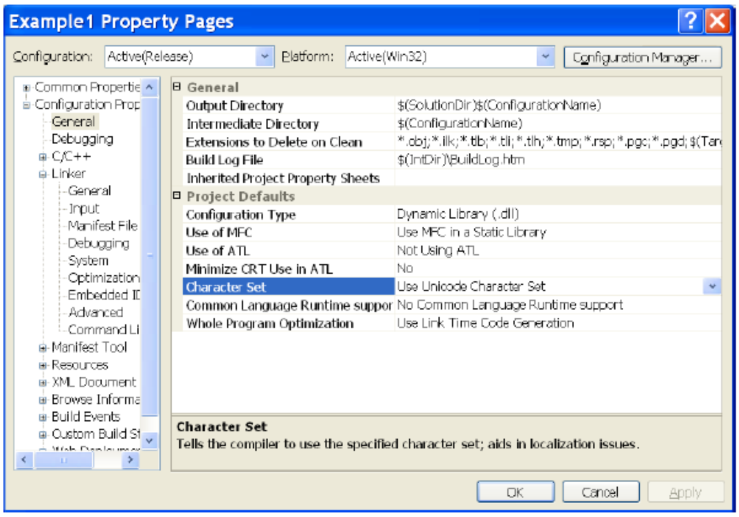
Solution Name: <Enter_name>

☑ Create directory for solution

□ Add to Source Control

Figure 10. MFC DLL creation in Microsoft Visual Studio. /10/

It is very important to do the definitions in properties right. Standard windows libraries need to be taken in use, e especially Unicode Character Set. These definitions are found in “PROJECT->PROPERTIES-> CONFIGURATION PROPERTIES -> GENERAL”. /10/



Example1 Property Pages

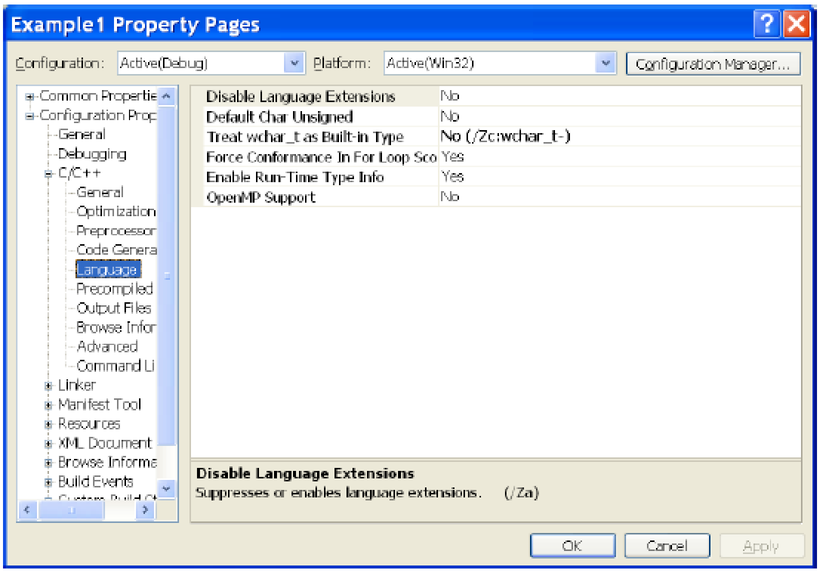
Configuration: Active(Release) Platform: Active(Win32)

General	
Output Directory	$(SolutionDir)$(ConfigurationName)
Intermediate Directory	$(ConfigurationName)
Extensions to Delete on Clean	*.obj;*.ilk;*.tlb;*.tli;*.tlh;*.tmp;*.rsp;*.pgc;*.pgd;$(Tar
Build Log File	$(IntDir)\BuildLog.htm
Inherited Project Property Sheets	
Project Defaults	
Configuration Type	Dynamic Library (.dll)
Use of MFC	Use MFC in a Static Library
Use of ATL	Not Using ATL
Minimize CRT Use in ATL	No
Character Set	Use Unicode Character Set
Common Language Runtime suppor	No Common Language Runtime support
Whole Program Optimization	Use Link Time Code Generation

Character Set
Tells the compiler to use the specified character set; aids in localization issues.

Figure 11. General Properties in Microsoft Visual Studio. /10/

The next step is go to “PROJECT->PROPERTIES -> C/C++ -> LANGUAGE”, where “Treat wchar_t as Built-in Type” must be set “No (/Zc:wchar_t-)”. After that go to tap “PREPROCESSOR” where the definition file (*.def) of the dll must be included in the pre-processor property. The name depends on the application name. /10/

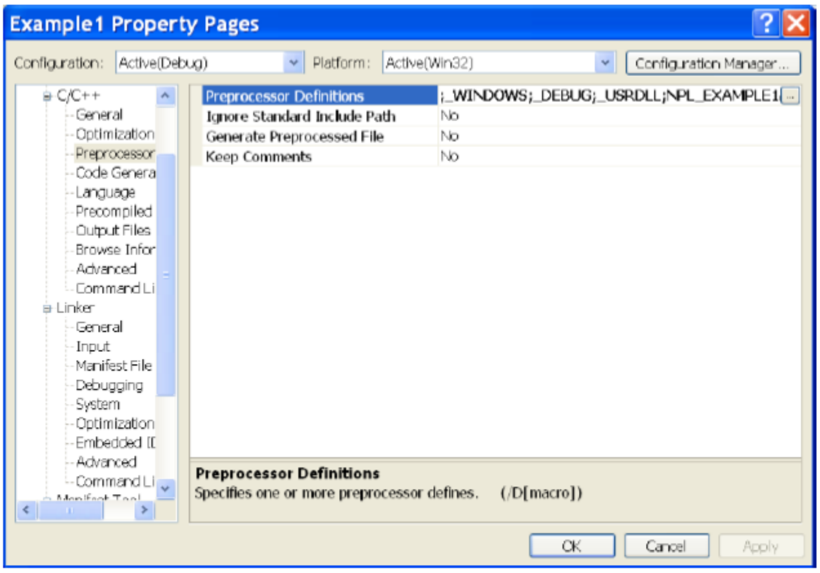


Configuration: Active(Debug) Platform: Active(Win32)

Disable Language Extensions	No
Default Char Unsigned	No
Treat wchar_t as Built-in Type	No (/Zc:wchar_t-)
Force Conformance In For Loop Sco	Yes
Enable Run-Time Type Info	Yes
OpenMP Support	No

Disable Language Extensions
Suppresses or enables language extensions. (/Za)

Figure 12. C++ Language settings in Microsoft Visual Studio. /10/



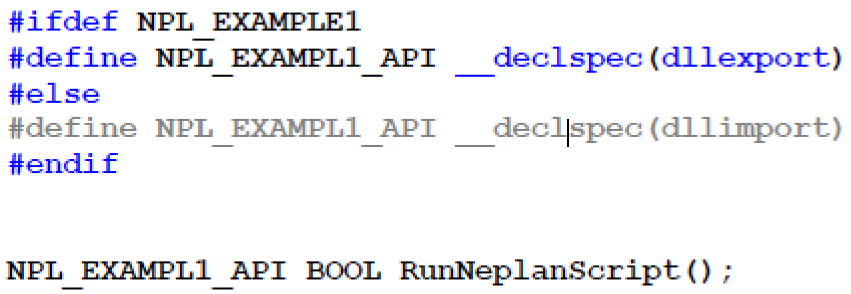
Configuration: Active(Debug) Platform: Active(Win32)

Preprocessor Definitions	;_WINDOWS;_DEBUG;_USRDLL;NPL_EXAMPLE1(
Ignore Standard Include Path	No
Generate Preprocessed File	No
Keep Comments	No

Preprocessor Definitions
Specifies one or more preprocessor defines. (/D[macro])

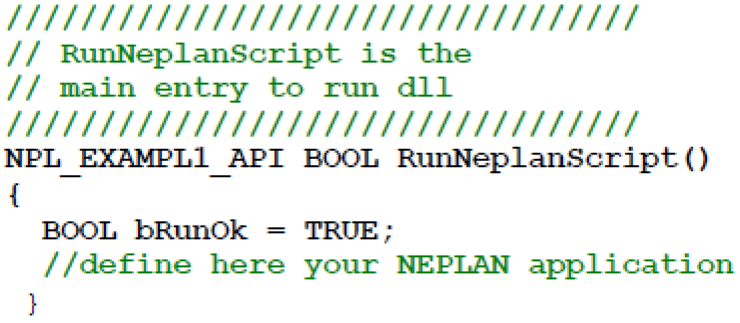
Figure 13. Pre-processors Definitions in Microsoft Visual Studio. /10/

In the header file of the main *cpp (e.g.”Example1.h”) file define the export function “RunNeplanScript as follows:



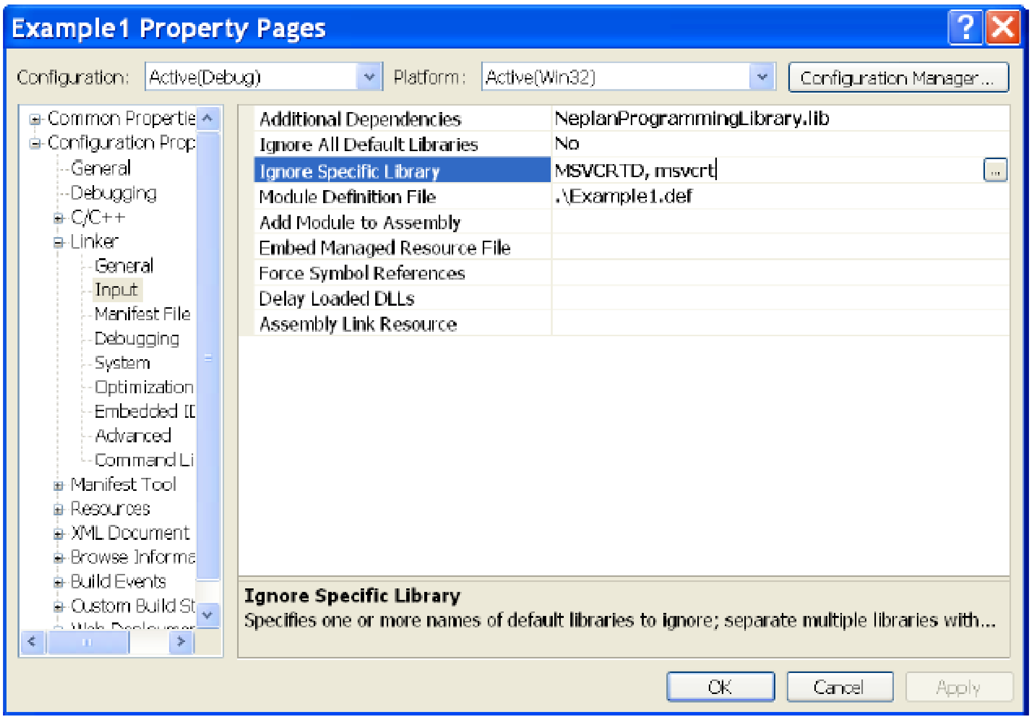
```
#ifdef NPL_EXAMPLE1
#define NPL_EXAMPL1_API __declspec(dllexport)
#else
#define NPL_EXAMPL1_API __declspec(dllimport)
#endif


NPL_EXAMPL1_API BOOL RunNeplanScript();
```

In the main *cpp file (e.g. Example1.cpp) write the “RunNeplanScript” as follows:



```
///////////////////////////////////
// RunNeplanScript is the
// main entry to run dll
///////////////////////////////////
NPL_EXAMPL1_API BOOL RunNeplanScript()
{
  BOOL bRunOk = TRUE;
  //define here your NEPLAN application
 }
```

Before the “NeplanProgrammingLibrary.lib” file was mentioned. That must be included in the project. Some specific libraries must be ignored at the same time. The ignored libraries are MSVCRTD and msvcrt. The place for ignoring is located in “PROJECT->PROPERTIES->LINKER->INPUT” where Ignore Specific Library needs to bechosen. /10/



Configuration: Active(Debug) Platform: Active(Win32)

Additional Dependencies	NeplanProgrammingLibrary.lib
Ignore All Default Libraries	No
Ignore Specific Library	MSVCRTD, msvcrt
Module Definition File	.\Example1.def
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded DLLs	
Assembly Link Resource	

Ignore Specific Library
Specifies one or more names of default libraries to ignore; separate multiple libraries with...

Figure 14. Ignore Specific Library in Microsoft Visual Studio. /10/

The last step is to open the module definition file in the Microsoft Visual Studio and there the export function "RunNeplanScript" must be defined.

```
; Example1.def : Declares the module parameters for the DLL.

LIBRARY      "Example1"

EXPORTS
    ; Explicit exports can go here
     RunNeplanScript   @1
```

Finally C++ programming can be started and after that the C++ program is done it is ready to be compiled into a dynamic link library which is DLL file. The file can be started as Neplan menu item “FILE->RUN NPL”. /10/

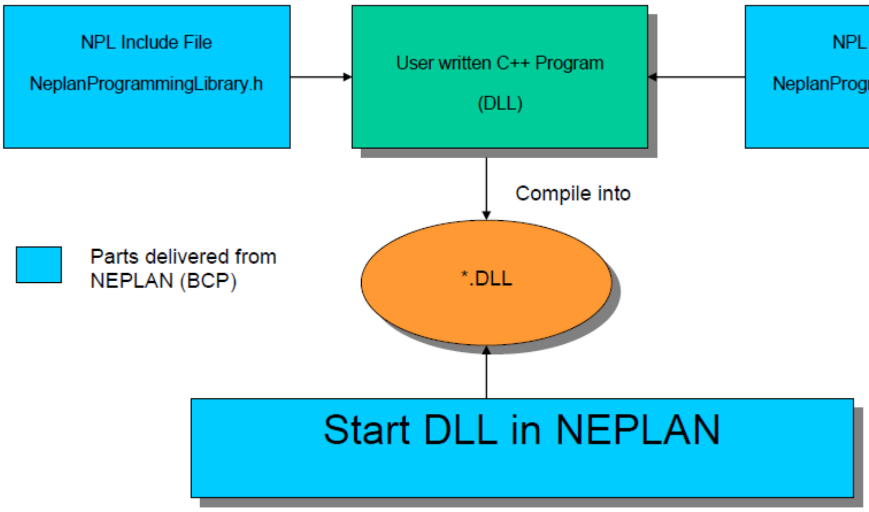


Figure 15. NPL Overview. /10/

7.1 NPL Application

NPL Application is programmed with Microsoft Visual Studio. In this case the application needs to read a row where is the keyword CTIMESTATE in the dynamic interface file. There is the date and time information when last control happened in the network because inside the NPL is an ImportSCADAFile function where the file name and the date and time of last control in the network have to be defined. All of them need to be in TCHAR form. TCHAR character type is Unicode which have to take account during a programming. With Unicode it is possible to use any language what user ever want. String class takes account only 256 characters which is enough when using western languages. Very important thing is to change the character set to Unicode in project settings when using Microsoft Visual Studio.

The date and time is needed to be read in C++ code. For that there is needed to use functions GETLINE and STRING STREAM. These functions works only with string class so there is needed to use conversion string to TCHAR. When the date and time is read in the dynamic interface file ImportSCADAFile function is ready to use in the code. That function will import control data in the dynamic interface file. After this load flow analysis can be started with RunAnalysisLF function. The code was built in the DLL file. This C++ program work like this:

1. Read the date and time in the dynamic interface file
2. Import dynamic interface file to Neplan or import data from SQL database to Neplan.
3. Run load flow calculation

All of this should work automatically. In the code it did with while function but it didn’t work in a correct way. If the script not get any value back then state of the network didn’t change but LF calculation running all the time and Neplan crashed. When the script get a value back then the state of network will change but in the same time program ends.

8 FUTURE POSSIBILITIES

8.1 Neplan 360

In future there is a need to investigate how to build an interface to work automatically. There are also other NA software in the market. Neplan offers Neplan 360 which works in the cloud or in the intranet. That could be a problem in the substation automation because usually only Local Area Network is built. The interface in Neplan 360 can be built with C# which is a more powerful programming language than C++ and today a more common programming language. The control and measurement data of the network is stored in the XML-files so MicroSCADA needs to write and read these files. There are functions to operate with text files but when operating with the XML files some intelligence needs to be coded with SCIL when operating with the XMLfiles.

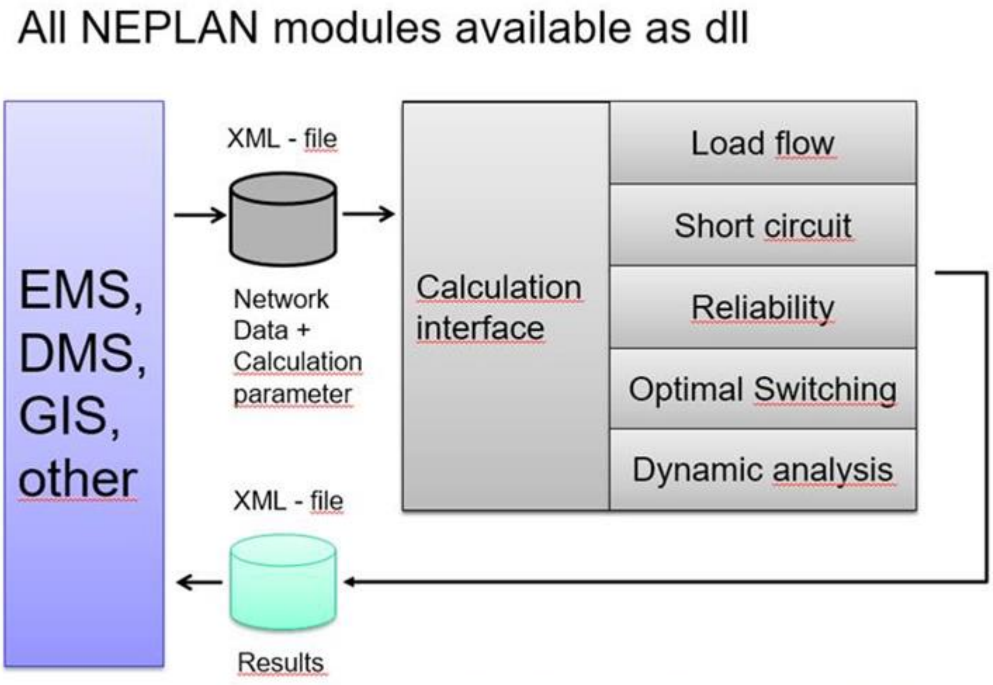


Figure 16. Interface system in Neplan 360.

8.2 Etap

One interesting NA software is Etap. It is also suitable for customer requirements when talking about analysis functions. Interface communication is practicable through the OPC protocol. The OPC protocol could be a really important protocol when transferring data between MicroSCADA and Etap. Etap needs more investigation in future because MicroSCADA supports the OPC protocol.

9 CONCLUSION

It seems that there is a demand to take NA software functions in use in MicroSCADA. Together SCADA and NA software can respond to the customer requirements. It is especially important to get the training mode t working in the future. Requirements need be taken into account carefully when planning this kind of system. It is obvious that this kind of system has a lot of requirements. More requirements means more work, especially when coding with C++. Testing the C++ code was quite difficult because every time when the C++ code changed, it needed to be compiled in the DLL file every time again and after that the DLL file needed to be run again in Neplan. In my opinion it is quite arduous work with the NPL API. With NPL API there was not a solution how to read the SQL database or the dynamic interface file automatically. For that reason practical working with MicroSCADA was kept in minimum. MicroSCADA can write and read data to text files which is really important when a text file works as an interface. Neplan 360 might resolve the problem of an automatic interface. Neplan 360 works in an internet or intranet which has to be taken into account. Other software that should be taken under the investigation is Etap. Etap has a possibility to use the OPC protocol to communicate with other systems. For that reason Etap is a potential NA software to integrate with MicroSCADA. MicroSCADA also supports the OPC protocol.

REFERENCES

/1/ ABB website. Accessed 30.3.2016 http://new.abb.com/about/abb-in-brief

/2/ ABB website. Accessed 30.3.2016 http://new.abb.com/fi/abb-lyhyesti/historia

/3/ ABB website. Accessed 30.3.2016 http://new.abb.com/fi/abb-lyhyesti/historia/suomalaiset-juuret

/4/ ABB_Suomessa_125_v.pdf. Accessed 11.4.2016

/5/ ABB website. Accessed 11.4.2016 http://new.abb.com/fi/uralle

/6/ Neplan website. Accessed 13.4.2016 http://www.neplan.ch/neplanproduct/en-electricity/

/7/ Neplan_E.pdf

/8/ NEPLAN-GIS-Interface-V30-2012.pdf

/9/ Neplan website. Accessed 16.4.2016 http://www.neplan.ch/wp-content/uploads/2015/01/NEPLAN-P-ResearchModules-engl.pdf

/10/ NPL - NEPLAN Programming Library Manual (Version 5.5.5)

/11/ http://www.sqlcourse.com/intro.html

/12/ Oracle website. Accessed 3.5.2016 https://docs.oracle.com/cd/B28359_01/server.111/b28286/intro002.htm

/13/ Ohjelmointiputka Accessed 10.5.2016 http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=cpp_ohj_04#tekstistring

/14/ ABB website. Accessed 11.5.2016 http://new.abb.com/fi/abb-lyhyesti/suomessa/yksikot/network-management

/15/ Customer specification. accessed 17.5.2016 NCP-00-E0-RP-5001.pdf

/16/ Etap website. Accessed 24.5.2016 http://etap.com/real-time/native-scada-communication-protocols.htm

/17/ MicroSCADA pro for substation automation. ABB