Laboratory Implementation of Automation Functions for Monitoring and Control

Module 5

MMS Mapping in IEC 61850

Contents

1. Introduction 3

2. Theory 3

2.1. Internet protocol suite 3

2.2. Distributed computing architectures 4

2.2.1. Client-Server 4

2.2.2. Peer-to-peer 5

3. Communication Services in IEC61850 5

3.1. Abstract Common Services Interface 5

3.2. Specific communication service mapping (SCSM) 8

4. MMS Protocol in IEC 61850 9

4.1. Functional Constraints 9

4.2. IEC 61850 MMS Mapping 9

4.3. libIEC61850 11

5. Student Tasks 11

5.1. Preparation 12

5.2. Laboratory exercieses 12

Bibliography 14

1. Introduction

In a communication network, information is exchanged through messages written according to communication protocols. A communication protocol is a set of rules and conventions that ensure common understanding of the involved parties –sending and receiving entities- on the meaning of the transmitted message.

Each communication protocol has established rules to describe a piece of information. Other protocols that describe the same information with different rules could be present or could still be designed though. In some system engineering practices, the designer has the flexibility to choose one communication protocol among numerous alternatives.

GOOSE and Sampled Value protocols were introduced in the earlier sessions of the 'Laboratory Implementation of Automation Functions for Monitoring and Control Laboratory'. The objective of this module is completing the vision of the students on the communication protocols used in power utility automation domain by introducing the protocol Manufacturing Message Specification (MMS).

This module provides an overview on the concepts necessary to understand the infrastructure and network architecture that enables information exchange between different IEDs (Chapter 2). Furthermore, it introduces the abstract modeling approach of IEC 61850 standard to describe communication services, which ensure interoperability in power utility automation despite the presence of different communication protocols (Chapter 3). The module contains practical assignments to be performed by using an open source tool, libIEC61850, which provides server and client libraries for IEC61850-MMS services.

2. Theory

Substation automation systems (SASs) monitor, protect and control the equipment in the substation. Traditionally, several point-to-point copper wires link the seperately located equipment. Advanced microprocessor technologies, which allow for implementation of automation functions quickly and reliably, transformed SAS into a digitized system with numerous computing units i.e. Intelligent Electronic Devices (IEDs). Therefore, the modern SASs are distributed computing systems that consists of several separately located electronics components. This chapter introduces the core communication concepts related to the distributed computing systems: internet protocol suite and network architectures.

2.1. Internet protocol suite

Billions of people all around the world use applications such as web, electronic mail and social media in their daily life. These essential services are provided through a globally interconnected computer network i.e. Internet.

Internet protocol suite provides the specifications for packing, addressing, transmitting, routing, and receiving the data in order to interconnect computers. The suite is a conceptual model with four abstraction layers (application, transport, internet and link). These layers classify all related protocols according to the scope of networking involved.

The layers of the protocol suite describes a hierarchy according to the closeness to the physical transmission of the data. Lower layers are closer to the physical components and higher layers are closer to the users. Definition of the layers in Internet protocol suite and some protocols reside in these layers are given in Table 1.

Table 1. Four layers of Internet Protocol Suite

Hierarchy	Layer	Function	Example
Lower to higher ↑	Application	Providing end-user applications and services	HTPP
	Transport	Handling host-to-host communication	TCP
	Internet	Device addressing, basic datagram communication and routing	IP
	Link	Interfacing the suite to the physical hardware on which it runs	Ethernet

2.2. Distributed computing architectures

A distributed computing system can be created through several network architectures depending on the environment, the priorities of the application and the available resources. This chapter provides a high-level information on two common class of network architectures and their strengths/flaws.

2.2.1. Client-Server

This architecture describes a network where the computer functions are divided between two different autonomous devices. Figure 3 shows the elements and interactions of client-server architecture.

- The client: the entity (can be a machine or a program) that makes a request to the server.
- The server: the entity (computer program) that provides functionalities and services to the client.

These two entities can be in the same computer or in different computers connected by a network. Following steps describe the typical information flow in a client-server architecture:

1. Client sends a data request to the server
2. Server accepts the request
3. Sever processes and packs the requested data
4. Server delivers the packets containing the requested data to the client

In principle server has the potential to serve multiple clients at the same time. However, this architecture can be implemented in instances with a single client connected to numerous servers simultaneously. This architecture does not allow direct communication between two clients. For client-client communication, the message has to pass through the server.

The centralized control makes the distributed computing secure and stable in client-server networks. Since the important data is stored on the server, data/file management is easy and data replication is avoided. In this sense, the client-server is a scalable architecture.

On the other hand, the user has to rely on a powerful device that performs memory and computing intensive operations. These are mostly expensive systems both at the installation and operation phases due to high quality hardware that requires expertise in technical details. Another disadvantage is that the failure of server would lead to failure in whole process.

2.2.2. Peer-to-peer

In the peer-to-peer architecture, the network consists of multiple connected entities that share resources without the central coordination of servers. In this architecture, peers are both suppliers and consumer of resources as opposed to the client-server model. Therefore, each node in the network can act as client and server as shown in Figure *2*.

Without the central administrative system, peer entities organize the access rights by setting sharing permissions on the resource to be shared. As peer-to-peer networks do not require an expensive dedicated hardware to function as a server, they are typically less expensive than client-server networks. With several servers in effect, peer-to-peer network accommodate less bottleneck risk for the server.

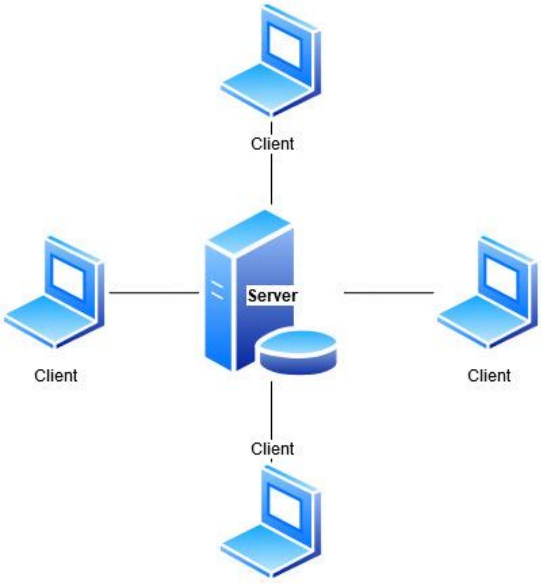


Figure 1. Client-server architecture

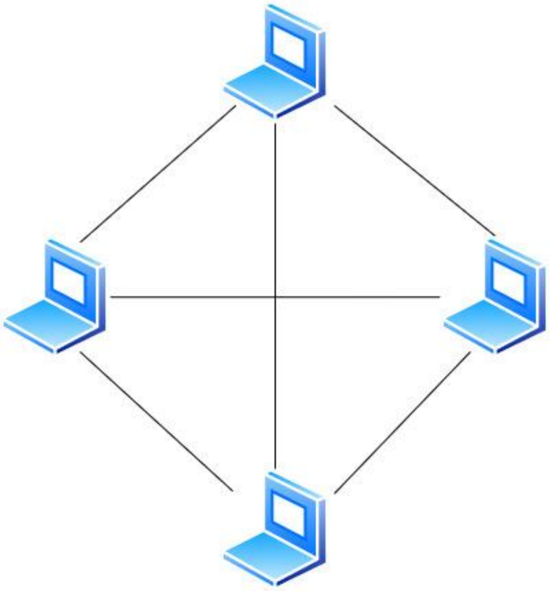


Figure 2. Peer-to-peer architecture

3. Communication Services in IEC61850

In substation automation context, communication services are the actions that enable cooperation between multiple entities in the system. Sending and receiving commands, reporting occurrence of some events, logging data are some examples of communication services.

IEC 61850 defines the communication services that a device can perform by using an object modeling technique. In this technique, services are defined by describing what (data, report, logging etc.) they provide in an abstract way. These abstract descriptions are mapped to particular communication protocols in order to encode concrete messages that contains the information to be exchanged are obtained.

3.1. Abstract Common Services Interface

Substation automation is a domain where the variety of communication protocols raises the interoperability challenge. In an SAS that consists of IEDs with vendor-specific communication protocols, the common understanding on the exchanged data could be achieved through complicated and costly protocol converters. However, such a system would be quite inflexible to alteration. Because in this case, substituting IEDs from different vendors would require making changes to the other elements' way of encoding messages in the system as well. A consensus on the way to exchange information between IEDs is necessary. IEC 61850 responds this need by abstraction of the communication services to be performed by IEDs.

Abstract Common Services Interface (ASCI) is the virtual interface to an IED providing abstract communication services such as connection, variable access, unsolicited data transfer, device control and file transfer services [1]. Abstraction decouples service modeling from the actual communication stack. Various mappings for different requirements can be done from the ASCI. Moreover, following state-of-the-art in communication technology is more convenient as the abstraction removes the need for altering the service model and, therefore, databases, etc. Figure 3 shows the model of information exchange between devices in ASCI.

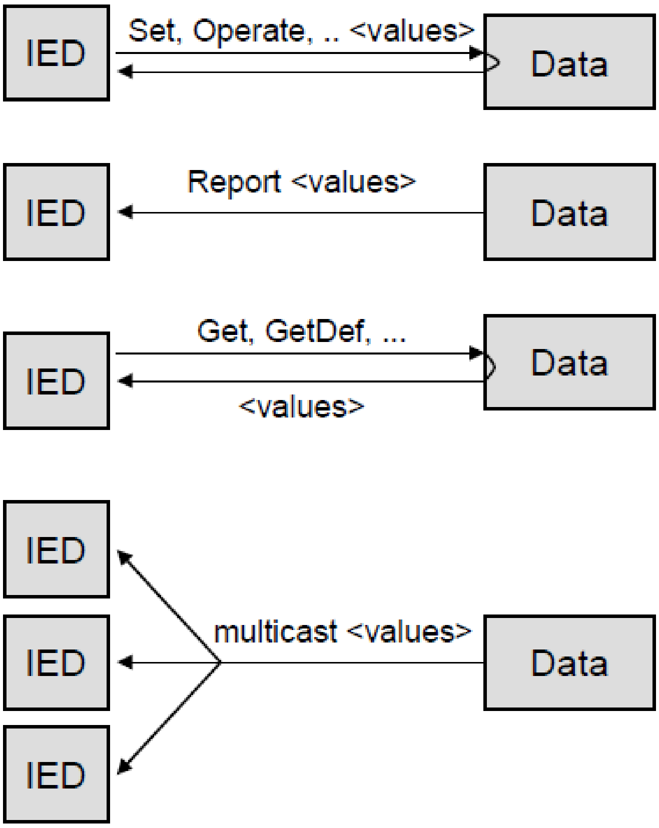


Figure 3. ASCI model for information flow [2]

Part 7-2 of the IEC 61850 standard defines the common utility services for both client-server and subscriber-publisher architectures. Services like set or get data values use the client-server architecture. On the other hand, GSE services (used for time-critical purposes, for example, fast and reliable transmission of data between protection IEDs, from one IED to many remote IEDs) and the sampled value services for transmissions based on a periodic basis use publisher-subscriber architecture. Some of the common services are listed in the Table 2.

Table 2. ASCI models and services [2]

Service model	Description	Services
Server	Represents the external visible behaviour of a device. All other ACSI models are part of the server.	GetServerDirectory
Application association	Provision of how two devices can be connected. Provides different views to a device: restricted access to the server's information and functions.	Associate Abort Release
Logical device	Represents a group of functions; each function is defined as a logical node.	GetLogicalDeviceDirectory
Logical node	Represents a specific function of the substation system, for example, overvoltage protection.	GetLogicalNodeDirectory GetAllDataValues
Data	Provides a means to specify typed information, for example, position of a switch with quality information, and timestamp.	GetDataValues SetDataValues GetDataDefinition GetDataDirectory
Data set	Allow to group various data together.	GetDataSetValues SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory
Reporting and logging	Describes the conditions for generating reports and logs based on parameters set by the client. Reports may be triggered by changes of process data values (for example, state change or deadband) or by quality changes. Logs can be queried for later retrieval. Reports may be sent immediately or deferred (buffered). Reports provide change-of-state and sequence-of-events information exchange.	Buffered RCB: Report GetBRCBValues SetBRCBValues Unbuffered RCB: Report GetURCBValues SetURCBValues Log CB: GetLCBValues SetLCBValues QueryLogByTime QueryLogAfter GetLogStatusValues
Control	Describes the services to control, for example, devices or parameter setting groups.	Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate

3.2. Specific communication service mapping (SCSM)

The abstract objects and services, which IEC 61850 defines in order to achieve interoperability, have to be mapped to an actual communication protocol in order to obtain the real-world instructions i.e. bits and bytes, which in fact carry out the communication services such as getting/setting data.

Specific communication service mapping (SCSM) defines the specific syntax (format), the way of encoding the messages and how the messages are passed through a network. IEC 61850 provides specifications for two types of SCSM [2]:

- IEC 61850-8-1: Mapping services to MMS and TCP/IP/Ethernet
- IEC 61850-9-2: Mapping directly on Ethernet

These two types aim at mapping the abstract services to different layers of the protocol suite. As can be seen in Figure *4*, IEC 61850-8-1 provides detailed instructions/specifications of protocols that operate over full OSI (7 layers of abstraction on the left) and Internet Protocol Suite (4 layers on the right of the figure). On the other hand, IEC 61850-9-2 specifications are on the protocols that directly access the link layer. The two SCSMs are suitable to different type of communication services in power system automation.

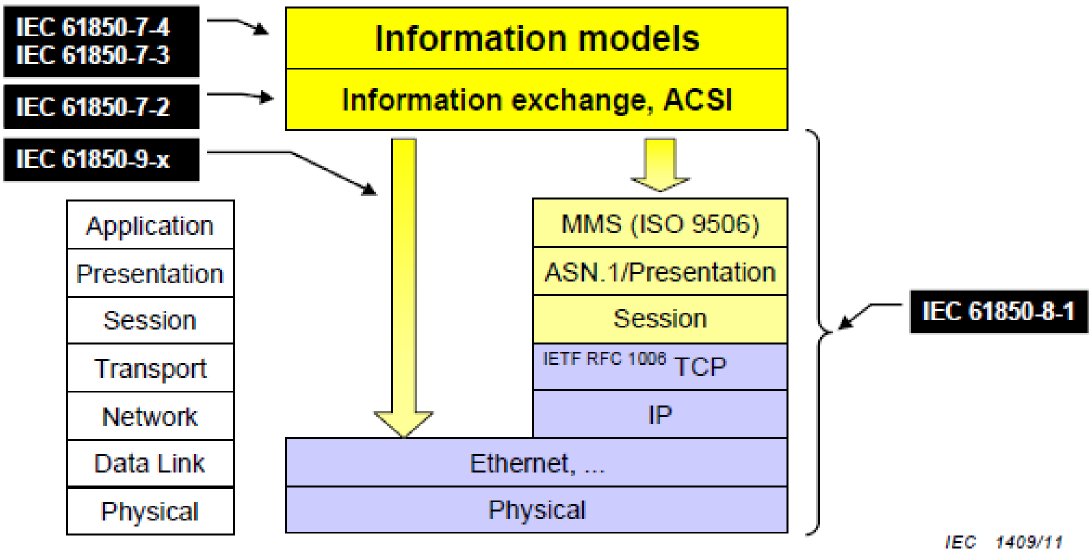


Figure 4. Example of communication mapping [2]

Direct mapping to Ethernet, which bypasses the transport and network layers, has the intrinsic advantage of speed as compared to the mapping through MMS and TCP/IP. This advantage makes this type of mapping more suitable to applications that require high-speed. Direct mapping is typically in the local domain of automation. Peer-to-peer architecture is feasible for this type of mapping since addressing/routing is not specified. Generic Object Oriented Substation Event (GOOSE) and Sampled Values (SV) are the communication protocols that use the direct mapping to Ethernet based on publisher-subscriber architecture of peer-to-peer network paradigm.

Mapping through MMS and TCP/IP offers a robust communication. Therefore, it is routable beyond the local domain and suitable to vertical information flow that requires less messaging speed.

4. MMS Protocol in IEC 61850

Manufacturing Message Specification (MMS) is a messaging system that defines rules for exchanging information between networked devices or computer applications about the real time conditions and supervisory control. MMS is defined by standards ISO/IEC 9506-1 (Services) and ISO/IEC 9506-2 (Protocol).

MMS uses an object-oriented modeling technique based on object classes that represent the units that interact with each other in a system. An object class is a set of attributes that characterizes the self of the object, and methods that characterizes the functions the object is involved in (such as like read, write, store, start, stop in communication domain). Accordingly, MMS defines standard objects that exist in devices that communicate with each other, standard messages exchanged within a client-server architecture to monitor and control these objects, and encoding these messages to bits and bytes.

4.1. Functional Constraints

IEC 61850 defines a typical automation function with a logical node. A logical node contains a list of data (for example, position) with dedicated data attributes. These attributes are classified according to their specific use such as controlling, reporting, logging, configuration, measurement. Functional constraints (FCs) are the properties that determine the use of the data attributes. They play a crucial role in the definition of the information models and in the services to access the various parts of the information model.

4.2. IEC 61850 MMS Mapping

The main object of MMS system is the Virtual Manufacturing Device (VMD). VMD is the server that makes available a set of resources and functionality associated with one or more devices for control, monitoring, or both purposes. An instance of an IEC 61850-7-2 Server class is mapped one-to-one to an MMS VMD object. Each VMD is assigned one or more communication addresses that create Service Access Points (SAPs). The SAPs allow for exchange of MMS services that manipulate MMS objects [3].

An MMS Domain object represents an instance of an IEC 61850-7-2 Logical Device. The server object contains one or more MMS Domain objects. An MMS Domain represents a collection of information associated with a specific name. For mapping to MMS, the Domain is used in order to represent the collection of objects and services that constitute a logical device [3] as shown in Figure 5.

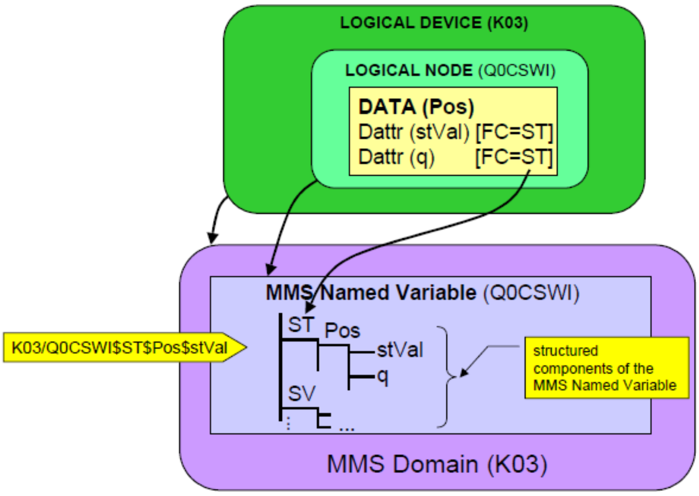


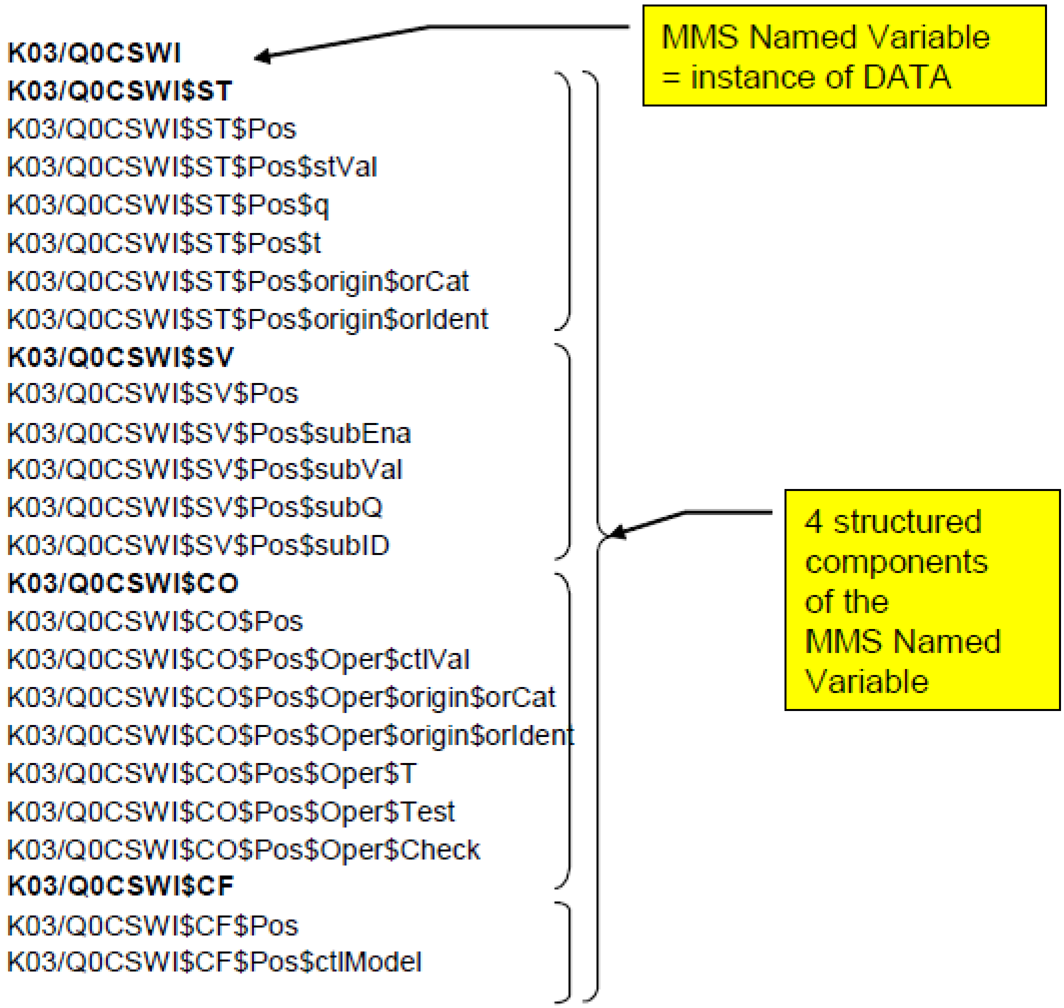
Figure 5. MMS mapping [2]

Each instance of an IEC 61850 Logical Node class maps to a single MMS Named Variable. The MMS Named Variable has a hierarchical MMS type description with multiple levels of components. In this hierarchy, the components of this named variable are constructed by selecting all data attributes with the same functional constraint (FC).

The data of an instance of Logical Node are decomposed into multiple MMS Named Components with separator '$'. The data attributes DataAttr of the DataObjects map in a similar manner to the DataObjects. However, the attribute name is also included within the hierarchy:

- <LNVariableName>$<FC>$<LNDataObjectName1>
- <LNVariableName>$<FC>$< LNDataObjectName1>$<AttributeName1>

Figure 6 shows an example of MMS decomposition:



```
K03/Q0CSWI
K03/Q0CSWI$ST
K03/Q0CSWI$ST$Pos
K03/Q0CSWI$ST$Pos$stVal
K03/Q0CSWI$ST$Pos$q
K03/Q0CSWI$ST$Pos$t
K03/Q0CSWI$ST$Pos$origin$orCat
K03/Q0CSWI$ST$Pos$origin$orIdent
K03/Q0CSWI$SV
K03/Q0CSWI$SV$Pos
K03/Q0CSWI$SV$Pos$subEna
K03/Q0CSWI$SV$Pos$subVal
K03/Q0CSWI$SV$Pos$subQ
K03/Q0CSWI$SV$Pos$subID
K03/Q0CSWI$CO
K03/Q0CSWI$CO$Pos
K03/Q0CSWI$CO$Pos$Oper$ctlVal
K03/Q0CSWI$CO$Pos$Oper$origin$orCat
K03/Q0CSWI$CO$Pos$Oper$origin$orIdent
K03/Q0CSWI$CO$Pos$Oper$T
K03/Q0CSWI$CO$Pos$Oper$Test
K03/Q0CSWI$CO$Pos$Oper$Check
K03/Q0CSWI$CF
K03/Q0CSWI$CF$Pos
K03/Q0CSWI$CF$Pos$ctlModel
```

MMS Named Variable = instance of DATA
4 structured components of the MMS Named Variable

IEC 1496/11

Figure 6. Example of MMS named variable (process values) [2]

ISO 9506 (MMS) specifies several MMS objects that could be used as part of this SCSM. However, not all objects are required in order to perform the mapping of IEC 61850-7-2, IEC 61850-7-3, and IEC 61850-7-4. Table 3 shows the MMS objects and services that shall be used within this SCSM [3].

Table 3. MMS objects and services in use within this SCSM [2]

MMS Object	IEC 61850 OBJECT	MMS SERVICES IN USE
Application Process VMD	Server	Initiate, Conclude, Abort, Reject, Cancel, Identify
Named Variable Objects	Logical Nodes and Data	Read, Write, InformationReport, GetVariableAccessAttribute, GetNameList
Named Variable List Objects	Data Sets	GetNamedVariableListAttributes, GetNameList, DefineNamedVariableList, DeleteNamedVariableList, Read, Write, InformationReport
Journal Objects	Logs	ReadJournal, InitializeJournal, GetNameList
Domain Objects	Logical Devices	GetNameList, GetDomainAttributes, StoreDomainContents
Files	Files	FileOpen, FileRead, ObtainFile, FileClose, FileDirectory, FileDelete

4.3. libIEC61850

libIEC61850 is an open-source (GPLv3) implementation of IEC 61850 client and server libraries. It is implemented in C language to provide maximum portability. It can be used to implement IEC 61850 compatible client and server applications on embedded systems and PCs running Linux and Windows [4].

The API (Application Programming Interface) of libIEC61850 provides client and server libraries for MMS and publisher and subscriber parts for GOOSE and SV (Sampled values) protocols. This API is intended to have a very low overhead and to be used to create very small IEC 61850 compliant server applications.

The server API is designed to be independent from the MMS mapping to be able to support other mappings (SCSM) in the future. Support for IEC 61850 includes the automatic generation of the MMS device model out of the IEC 61850 data model. The client API is a high-level API for implementing IEC 61850 compliant clients. Model discovery, variable read/write, reporting and control services are supported by this library [5].

5. Student Tasks

This session aims at showing the workflow of the communication services implemented via MMS protocol. The attendees will implement basic automation functions that involve two computing entities that reside in different ends in a client-server architecture. These functions will be performed by using the functions of the corresponding API of the library libIEC61850.

Next chapters provide the elements of preparation that is required to accomplish the tasks in the exercise session and the steps of the tasks in details.

5.1. Preparation

Participants of the practical session should review the following concepts in advance:

1. Network architectures
   1.1. Server-Client
   1.2. Publisher-Subscriber
2. IEC 61850
   2.1. Configuration files
   2.2. Logical Device, Logical Node, Data Object, Data Attribute, Data Set
   2.3. ASCI
   2.4. Mapping

The tool with which the practical assignments of this session will be done is a C based API. Therefore, basic skills in C programming are required to complete the assignments. Knowledge and experience on the following C concepts are prerequisite for all participants:

1. Review basic syntax rules of C programming
   5.2. Header files
   5.2. Main function
   5.2. Tokens in C (Keywords, Identifiers/Variables, Constants, Punctuators, Operators)
2. Functions
3. LinkedList sturctures

Upon request, the instructors can provide tutorials for preparation of the participants with no background in advance. Note that practical pre-work is strongly recommended also for experienced C programmers to brush up on the required skills to accomplish the assignments.

5.2. Laboratory exercieses

At the beginning of the exercise session, the instructors will provide two template scripts on which the participant will implement server and client functions of libIEC61850. To minimize the programming effort, the template code will include some comment lines that indicate the position necessary C code needs to be written to.

The following list shows the sub-sessions of the practical session and elaborates the steps of the exercise. The required libIEC61850 functions for each step are attached to the corresponding exercise steps. The documentation for these functions will be provided in the session. Participants can also access libIEC61850 API documentation before the session [4].

1. **Introduction to libIEC61850 Server and Client APIs**
   1.1. Write a connection handler function that displays a log in server terminal in case of client connection
   1.2. Create an IedServer object for the given IEC model description (static_model.c)
   Required functions: *IedServer_create*
   1.3. Add the connection handler written in step 1.1 to IedServer
   Required functions: *IedServer_setConnectionIndicationHandler*
   1.4. Instruct IedServer object to listen to the connections
   Required functions: *IedServer_start*
   1.5. Create a IedClient object
   Required functions: *IedConnection_create*
   1.6. Connect the client to the server initialized in 1.2
   Required functions: *IedConnection_connect*
   1.7. See the connection log message in the server terminal

2. Information model of IEC 61850

2.1. View the CID file to see the logical nodes (LNs) of the IED

2.2. Interpret the data objects (DOs) of the LN class MMXU in IEC 61850-7-4

2.3. Print the information model of the IED according to the IEC 61850 7-2 in the client terminal
Required functions: *IedConnection_getLogicalDeviceList, IedConnection_getLogicalDeviceDirectory, IedConnection_getLogicalNodeDirectory*

```
For each LD in LogicalDeviceList:
        Print ID of LD
        For each LN in LogicalNodeList (of LD):
                Print LN
                For each DO in DataObjectList (of LN):
                        Print ID
                End For
        End For
End For
```

Figure 7. Algorithm to print information model of IED according to IEC 61850 7-2

3. MMS Mapping

3.1. Write down the information model of the IED according to the structure of MMS

3.2. Print the information model of the IED according to MMS in the client terminal
Required functions: *IedConnection_getLogicalDeviceVariables, IedConnection_getMmsConnection, MmsConnection_readVariable, MmsValue_toFloat*

3.3. Interpret the model difference between IEC 61850-7-4 and MMS models

4. Reading analog measurements from the remote server

4.1. Update the magnitude of the total active power of the IED measurement such that ‘MMXU2.TotW.mag.f=100 Watt’
Required functions: *IedServer_updateAttributeValue*

4.2. Read MMS value for the updated analog measurement by client request
Required functions: *MmsConnection_readVariable, MmsValue_toFloat*

5. Server datasets

5.1. View CID file to see the datasets

5.2. Read the name of all server datasets by client request
Required functions: *IedConnection_getLogicalNodeDirectory*

5.3. Create a server dataset by client command. The dataset should have several items including ‘MMXU2.TotW.mag.f[MX]’
Required functions: *IedConnection_createDataSet*

5.4. Read the dataset values
Required functions: *IedConnection_readDataSetValues, ClientDataSet_getValues*

Bibliography

[1] International Electrotechnical Commission, 'IEC Glossary'

[2] *Communication networks and systems for power utility automation, Part 7-1: Basic communication structure- Principles and models*, EN 61850-7-1:2011

[3] *Communication networks and systems for power utility automation, Part 8-1: Specific communication service mapping (SCSM) -Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*, EN 61850-8-1:2011

[4] libIEC61850 API Reference Manual. [Online]. Available: https://support.mz-automation.de/doc/libiec61850/c/latest/index.html

[5] MZ Automation GmbH., 'Open source libraries for IEC61850 and IEC 60870-5-104', [Online]. Available: https://libiec61850.com/libiec61850