Paper accepted for presentation at 2009 IEEE Bucharest Power Tech Conference, June 28th - July 2nd, Bucharest, Romania

Fault Data Collection in Substations According to IEC 61850

S. Mesentean, H.Frank, *Member, IEEE*, K. Fleischmann, and M. Stuhler

***Abstract*--In a research project a fault data collection device for substations with a communication interface according to IEC 61850 was developed. The IEC 61850 standard is the future of substation automation networking. This standard has no models for signaling faults included up to date. Therefore, specific logical nodes were defined by using attribute types of the standard. These additional nodes were implemented and tested in a self made server.**

***Index Terms*--Fault data collection, IEC 61850, substations.**

I. INTRODUCTION

In electrical power grids a very high operational availability of the equipment, subsystems and the whole grid is one of the most important requirements. If a fault occurs in a device it often causes subsequent faults. To enable a fast and an efficient diagnostic, fault data collection devices are implemented all over the power grid, which collect individual faults and store them together with a time stamp. The faults are signalled with flashing lights directly on the devices and they are reported via a communication network to a central SCADA system (SCADA – supervisory control and data acquisition). So a fast and efficient signalling, diagnostic and repair is possible.

For the communication networks in the field of power engineering during the last years a new standard was developed. The IEC 61850 “Communication Networks and Systems in Substations” is the standard protocol which meets the needs of the information flow [1]. With IEC 61850 the following main features can be achieved:

- It enables interoperability between devices from different manufacturers.
- The different levels configuration of the communication interfaces allows the implementation of many different automation functions.
- The structure of the standard can be adapted in future to new technologies in communication networks. Therewith long term stability is achieved.

The IEC 61850 standard supports data modelling and communication based on Ethernet systems. It reduces setup and configuration, and has greater performance overall [2]. It provides different data models for elementary functions in power engineering devices [3], [4]. They are named as logical nodes. With such logical nodes a model can be built up for the automation functions from devices in a subsystem of the electrical power grid. For the communication between the automation functions ACSI-services are standardized (ACSI – abstract communication service interface). These ACSI services are then mapped to real communication protocols.

In the following a fault data collection device is described which was equipped with a communication interface according to IEC 61850.

II. METHODS FOR SIGNALING FAULTS

Different sequences for signalling faults are specified in DIN 19235 [3]. Out of this, the following signalling types are supported by our device:

- fault signal with permanent light (DIN 19235 4.1.1),
- new value signal with simple or double flashing light (DIN 19235 4.1.2.1, ANSI/ISA-S18.1),
- initial value signal (DIN 19235 4.1.3.1).

In Fig. 1 the functional diagram of a new value signal with double flashing light can be seen. In this diagram every new operating state is indicated by a flashing signal. The central and local signals are switched on at the same time (fast flashing frequency). The flashing frequency will become slower after the acknowledgement of the fault, and will be completely switched off after cancellation (done by the delete button). If the operating state to be signalled is still available after acknowledgement, the flashing light switches to a permanently on indication. This will go to a lower flashing frequency only after the operating state changes over to the non-signalled state. The flashing light will then disappear after it is deleted (different button as the acknowledge one).

The difference between the new value signal with flashing light and the fault signal with permanent light consists in the fact that the operating state to be signalled is indicated directly by a permanent light. At the same time, as in the previous case, the central and the local indicator are switched on. The central audible signal and the local visual signal are switched off after the same principle presented in the previous case.

This work was supported in part by the German Federal Ministry of Economics and Technology (PROINNO II).

S. Mesentean is with Reinhold-Würth-University of the Heilbronn University, 74653 Künzelsau, Germany (e-mail: mesentean@hs-heilbronn.de).

H. Frank is with Reinhold-Würth-University of the Heilbronn University, 74653 Künzelsau, Germany (e-mail: frank@hs-heilbronn.de).

K. Fleischmann is with Unitro Fleischmann, 71522 Backnang, Germany (e-mail: k.fleischmann@unitro.de).

M. Stuhler is with Unitro Fleischmann, 71522 Backnang, Germany (e-mail: M.Stuhler@unitro.de).

978-1-4244-2235-7/09/$25.00 ©2009 IEEE

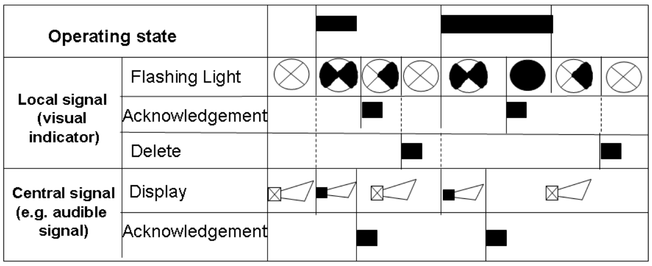


Fig. 1. Function diagram: *New value signal with double flashing light*.

Another case is represented by the initial value signal which is a function expansion of the new value signal. Only the first operating state to signal will flash, all subsequent signals will be assigned to the permanent light. At the same time one or more signals of the assigned central acoustic and local visual indicator are switched on. The acoustic indicator is switched off immediately when acknowledged. Flashing remains active until the acknowledge signal is acknowledged. This second acknowledge causes the flashing light to switch to a permanent light. At the same time, the acoustic alarm is reset. If the operating state to be signalled is still available during acknowledgement, the visual indicator remains activated as a "permanent light". It switches off only after the operating state transfers to the non-signalled state. For more details see [5].

In our case the signal will be a new value signal with double flashing light.

III. FAULT DATA COLLECTION DEVICE ACCORDING TO IEC 61850

The fault data collection device is shown in Fig. 2. It accomplishes high EMC ratings (EMC-electromagnetic compatibility) especially for power station applications. Fault-signals can be connected to this device as binary switch-signals. The states of the faults are signalled directly with flashing lights on the device (shown by the red lights in Fig. 2) according to the methods described in section II. They also can be transmitted through communication services according to IEC 61850 to other intelligent electronic devices (IED) or even to SCADA-systems.

The fault data collection device includes also an alarm state storage for the case of power failure. It can be programmed via an USB-port.

The internal structure of the fault data collection device (Server) is shown in Fig. 3. It includes the following main-components: a mapping-software, a data file, a communication model, a name list file, communication services and a system clock.

The mapping-software reads the status of the fault signals and stores them together with time-stamps and quality-information into a data-file. This program can also be used for reporting a fault or a change in data provided by the fault signals.

The data file includes for each fault-signal a logical node (LN). Each LN contains general information (common to all LN), status information, event controls and configuration data.

The communication interface is described by a model in a so called SCL-File (SCL – substation control language). This file specifies the structure of the data-file and the communication-services, which are supported by the fault data collection device (see Fig. 4). The information content from the SCL has to be mirrored into the data file.

Fig. 2. Fault data collection device (ME16-P USB).

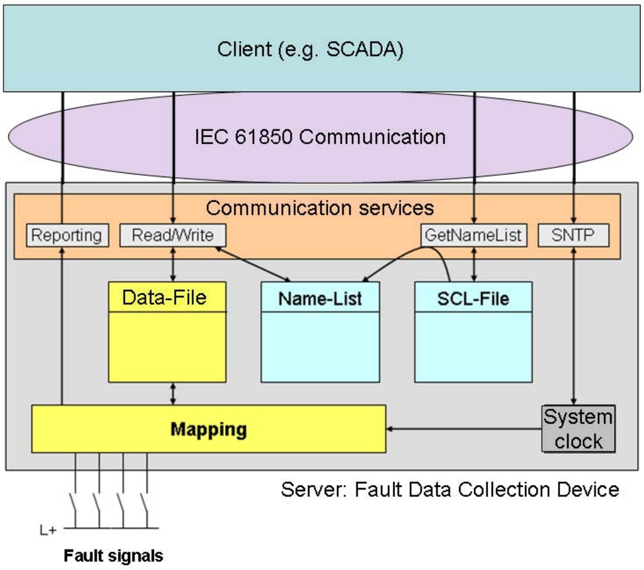


Fig. 3. Internal structure of the fault data collection device.

The name-list file is mainly used by the communication services in order to get a quick and comfortable overview about the structure of the data object elements. This architecture is based on the SCL (more details about the name-list see the communication chapter part B).

The IEC 61850 standard defines a big variety of communication services which can be used for the communication between the client and the server. Not all of them are equally used. The services used by the fault data collection device are the GetDirectory, GetDataObjectDefinition, Read/Write and ReportSettings. The GetDirectory and the GetDataObjectDefintion are used in an initialization to transmit the model of the communication interface from the server to the client. The other services are used for the communication when the device is in operation.

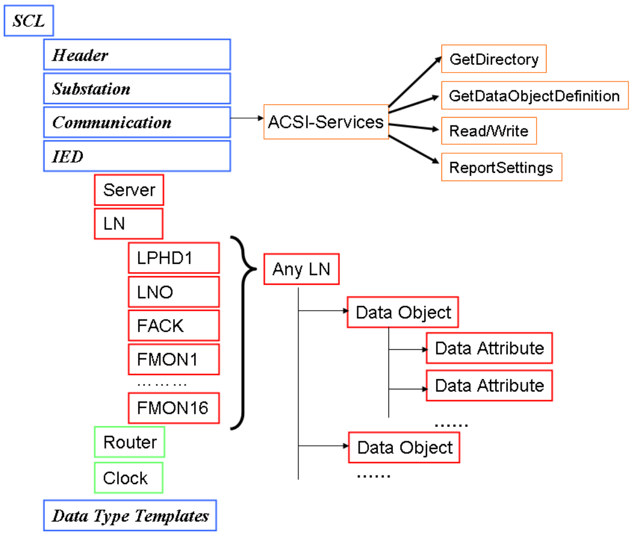


Fig. 4 Detailed schema of an SCL.

In order to get accurate time stamps for the fault signals, the device has a system clock. For the synchronization of this clock with a master clock the IEC 61850 specifies the SNTP-protocol (SNTP-simple network time protocol). With this protocol in general an accuracy of better than 1 ms can be achieved.

IV. MODEL OF THE COMMUNICATION INTERFACE

Due to the fact that the existing standard does not contain logical nodes representing information needed for signaling faults (for details see [1]-part 7-4), two new specific logical node-classes (FMON – fault monitoring and FACK – fault acknowledgement) were defined by using attribute types of the standard (see Fig. 5 and Fig. 6).

For each binary fault signal, which is connected to the fault data collection device, the data attributes corresponding to a FMON-logical node have to be implemented in the data file (Fig. 5). Its most important data object is the *IndivFault* which is mandatory. It contains the information whether a fault is active or not and it is structured according to the SPS-common data class (SPS – single point status). This data class includes a data attribute *stVal* for the status value, an attribute *q* for the quality of the status value and an attribute *t* which includes the time-stamp for the last change of the status value. For the status of the flashing light of the fault signal three optional data classes "*IndicPermanent*", "*IndicHighFreq*" and "*IndicLowFreq*" can be implemented. Each common data class used contains a series of data attributes. Some of this data attributes are mandatory some are optional and some can be mandatory or optional if certain conditions are fulfilled (for more details see [1]-part 7-3).

FMON class				
Attribute Name	**Attr. Type**	**Explanation**	**T**	**M/ O**
LNName				
Data				
Common Logical Node Information				
		LN shall inherit all Mandatory Data from Common Logical Node Class		M
Status Information				
IndivFault	SPS	FALSE: No individual fault TRUE: Individual fault		M
IndicPermanent	SPS	FALSE: Individual indicator (light) not permanent on TRUE: Individual indicator (light) permanent on		O
IndicHighFreq	SPS	FALSE: Individual indicator is not flushing with high frequency TRUE: Individ. indicator is flushing with high freq.		O
IndicLowFreq	SPS	FALSE: Individual indicator is not flushing with low frequency TRUE: Individ. Indicator is flushing with low freq.		O
Configuration				
ClOpCircuit	SPS	FALSE: closed circuit signal TRUE: open circuit signal		O
RespDelay	MV	Response delay in µs		O
FaultGroup	INS			O
OpMsg	SPS	FALSE: fault message TRUE: operational message only		O
DisableMsg (1)	SPS	FALSE: not disabled TRUE: disabled		O
OutputNo	INS	Output No … = Value of stVal		O
FaultDescript	INS	Description of the fault		O

Fig. 5. FMON-class (Fault-Monitoring) for signaling faults

FACK class				
Attribute Name	**Attr. Type**	**Explanation**	**T**	**M/O**
LNName				
Data				
Common Logical Node Information				
		LN shall inherit all Mandatory Data from Common Logical Node Class		M
Control Value				
AlarmAck	SPC	Alarm Acknowledgement		M
Clear	SPC	Clear		M
AudAck	SPC	Audible Acknowledgement		M
Status Information				
FaultGroup	INS			O
OpticalSignal	SPS	FALSE: OFF; TRUE: ON		O
AudibleSignalGr1	SPS	FALSE: OFF; TRUE: ON		O
Configuration				
MonitMode	SPS	False: "New Value Signal" True: "Individual Value Signal"		M
FlushMode	SPS	False: One Frequency True: Two Frequencies		M

Fig. 6. FACK-class (Fault-Acknowledgement) for the acknowledgement of faults

The FMON-LN-class supports also the implementation of configuration values. In the “*ClOpCircuit*”-data object it can be configured whether a fault-signal is realized as open or closed circuit. In the “*RespDelay*”-data object a delay time for reading a fault signal can be configured. In the “*FaultGroup*” data class a fault signal can be associated to a fault group.

For the acknowledgement of faults from a SCADA-system for each fault group a common FACK-logical node is provided (Fig. 6). The corresponding data attributes must also be implemented in the data file. Its control data objects are mandatory and represent the acknowledgement of the local-visual (*AlarmAck*) and central-audible signal (*AudAck*) together with the *Clear* command necessary to make the flashing light disappear. Other mandatory data objects are the configuration values “*MonitMode*” responsible for the type of the value signal (if new or individual) and “*FlushMode*” which shows if the frequency is simple or double. Responsible for the status information are the optional “*FaultGroup*”, “*OpticalSignal*” and “*AudibleSignalGr.1*” data objects. They can inform about the presence/absence of a certain type of signal (fault, optical, audible).

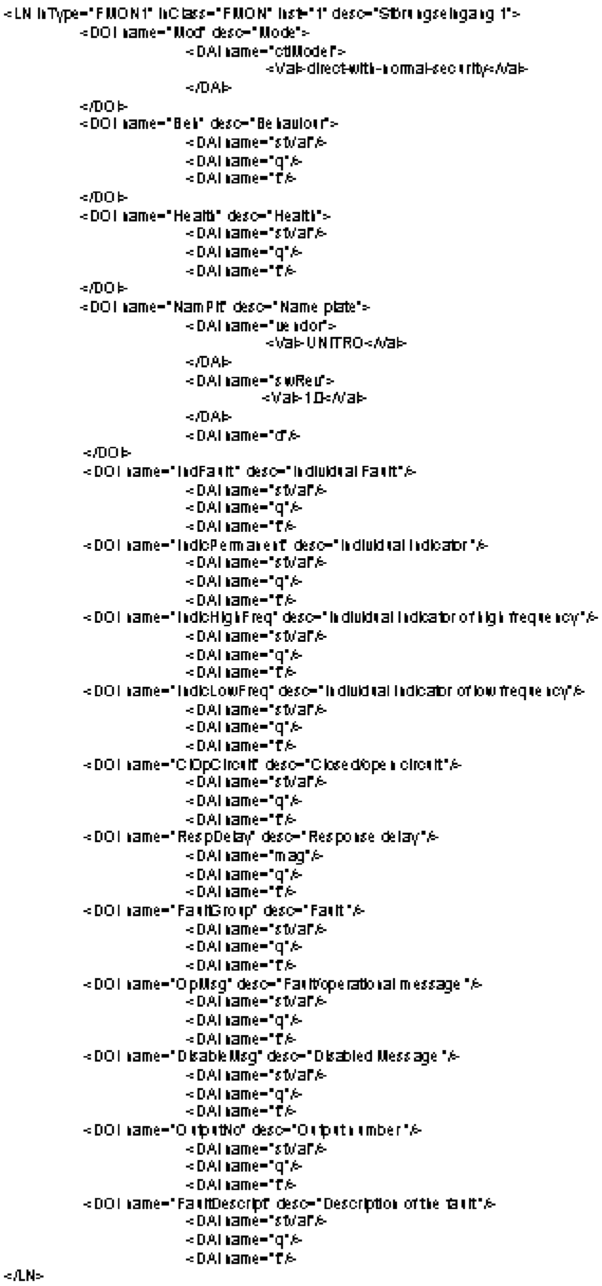
As already mentioned above, the model of the communication interface of the fault data collection device has to be described in a SCL-file. This file is an XML (eXtensible Markup Language) based file format which can be used to exchange configuration information between IEDs [6]. In this form the configuration information is available in a standardized file format. Fig. 7 shows as an example the model for one fault signal (FMON1).

V. COMMUNICATION

A. General Overview.

The communication services in the fault data collection device are specified as so called ACSI-services (ACSI - Abstract Communication Service Interface). These services are realized as MMS-protocol (MMS – manufacturing message specification) on Ethernet-TCP/IP. MMS defines a structure for the messages required to control and monitor the devices; it is not concerned with the way the messages are transferred between devices over the network.

The communication starts with an initialization where general information about the connection and the model of the communication interface in the fault data collection device (server) is transmitted to a client (see Fig. 8). After the initialization, when the communication is in operation, the client can read and write data elements in the server. When the server recognizes a fault, it uses the reporting-service to transmit an event message to the client. The protocol stack in the fault data collection system was realized by a self made program named IEC-Server Protocol Stack.



```
<LN lnType="FMON1" lnClass="FMON" inst="1" desc="Störungseingang 1">
        <DOI name="Mod" desc="Mode">
                <DAI name="ctlModel">
                        <Val>direct-with-normal-security</Val>
                </DAI>
        </DOI>
        <DOI name="Beh" desc="Behaviour">
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        </DOI>
        <DOI name="Health" desc="Health">
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        </DOI>
        <DOI name="NamPlt" desc="Name plate">
                <DAI name="vendor">
                        <Val>UNITRO</Val>
                </DAI>
                <DAI name="swRev">
                       <Val>1.0</Val>
                </DAI>
                <DAI name="d"/>
        </DOI>
        <DOI name="IndFault" desc="Individual Fault"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="IndicPermanent" desc="Individual Indicator"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="IndicHighFreq" desc="Individual Indicator of high frequency"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="IndicLowFreq" desc="Individual Indicator of low frequency"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="ClOpCircuit" desc="Closed/open circuit"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="RespDelay" desc="Response delay"/>
                <DAI name="mag"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="FaultGroup" desc="Fault"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="OpMsg" desc="Fault/operational message"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="DisableMsg" desc="Disabled Message"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="OutputNo" desc="Output number"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
        <DOI name="FaultDescript" desc="Description of the fault"/>
                <DAI name="stVal"/>
                <DAI name="q"/>
                <DAI name="t"/>
</LN>
```

Fig. 7 Piece of the SCL representing the FMON1 logical node

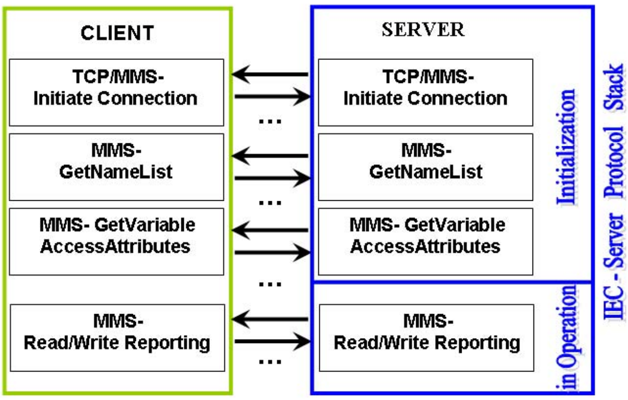


Fig. 8. IEC 61850 Communication.

B. *Initialization*

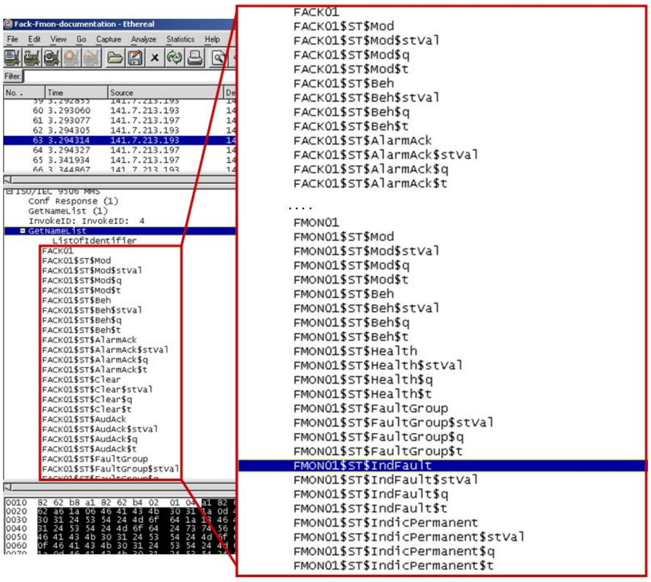
At the beginning a TCP connection is established between a client and the server.

After that the GetDirectory-ACSI-service is used to retrieve a list of the logical nodes, data objects and data attributes in the server. This ACSI-service is implemented in the MMS-protocol by using the GetNameList-telegrams. Fig. 9 shows an example for a GetNameList telegram.

When the client knows the names of all data elements in the server, it asks in the next step for their data types. For that it uses the GetDataObjectDefinition-ACSI-service. It is implemented in the MMS-protocol by using the GetVariableAccesAttributes-telegrams.

Another ACSI-service which is used in the initiation part is Read. It is used for transmitting the values each data element has, to the client.

The fault data collection device includes a communication interface with 17 LN with 793 data elements. To transmit the model for this communication interface from the server to the client about 3850 MMS-telegrams are required.



```
 60 3.293060   141.7.213.193
 61 3.293077   141.7.213.197
 62 3.294305   141.7.213.193
 63 3.294314   141.7.213.193
 64 3.294327   141.7.213.197
 65 3.341934   141.7.213.197
```

```
ISO/IEC 9506 MMS
   Conf Response (1)
   GetNameList (1)
   InvokeID: InvokeID:  4
 GetNameList
      ListofIdentifier
FACK01
FACK01$ST$Mod
FACK01$ST$Mod$stVal
FACK01$ST$Mod$q
FACK01$ST$Mod$t
FACK01$ST$Beh
FACK01$ST$Beh$stVal
FACK01$ST$Beh$q
FACK01$ST$Beh$t
FACK01$ST$AlarmAck
FACK01$ST$AlarmAck$stVal
FACK01$ST$AlarmAck$q
FACK01$ST$AlarmAck$t
. . . .
FMON01
FMON01$ST$Mod
FMON01$ST$Mod$stVal
FMON01$ST$Mod$q
FMON01$ST$Mod$t
FMON01$ST$Beh
FMON01$ST$Beh$stVal
FMON01$ST$Beh$q
FMON01$ST$Beh$t
FMON01$ST$Health
FMON01$ST$Health$stVal
FMON01$ST$Health$q
FMON01$ST$Health$t
FMON01$ST$FaultGroup
FMON01$ST$FaultGroup$stVal
FMON01$ST$FaultGroup$q
FMON01$ST$FaultGroup$t
FMON01$ST$IndFault
FMON01$ST$IndFault$stVal
FMON01$ST$IndFault$q
FMON01$ST$IndFault$t
FMON01$ST$IndicPermanent
FMON01$ST$IndicPermanent$stVal
FMON01$ST$IndicPermanent$q
FMON01$ST$IndicPermanent$t
```

```
0010  82 62 b8 a1 82 62 b4 02  01 04 a1 82
0020  62 a6 1a 06 46 41 43 4b  30 31 1a 0d
0030  30 31 24 53 54 24 4d 6f  64 1a 13 46
0040  31 24 53 54 24 4d 6f 64  24 73 74 56
0050  46 41 43 4b 30 31 24 53  54 24 4d 6f
0060  0f 46 41 43 4b 30 31 24  53 54 24 4d
```

Fig. 9 The GetNameList MMS message corresponding to the list with all the variables our model has.

C. *Operation*

When the communication is in operation the following functions can be performed between client and server over the communication network:

- The client can read data attributes corresponding to each data object but a special interest represent the ones from status information,
- The client is able to write data attributes into the server especially the configuration data.
- When a fault signal is recognized by the server it can send an event message to the client (reporting).
- The client can acknowledge fault messages.

The sequence for the transmission of fault messages and acknowledgement is shown as an example in Fig. 10. At the beginning there is no fault and therefore the operating state of the fault signal is FALSE and the flashing light is OFF. When a fault appears (the fault signal is TRUE), the display shows a fast flashing signal and the LN corresponding to that specific fault (here FMON number X) reports this event to the client. When an operator acknowledges this fault, the client sends a Write-telegram back to the server. If the fault is still persisting, then the display will show a permanent light. When the fault is repaired the flashing frequency will become slower and the event will be then reported from the server to the client (fault signal is FALSE). After the operator deletes the fault indication, the fault display will then change again to the OFF-state.

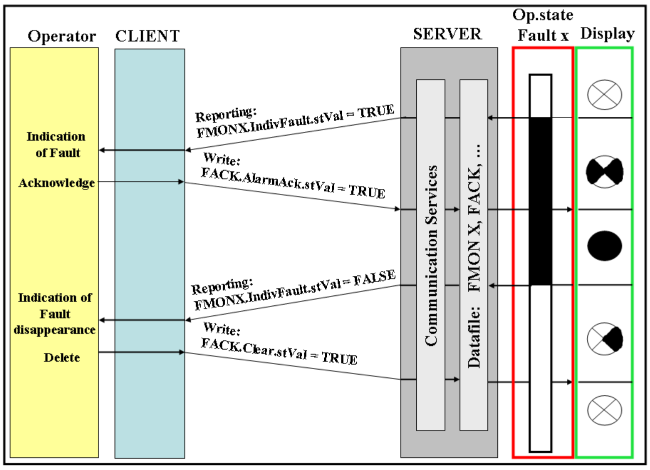
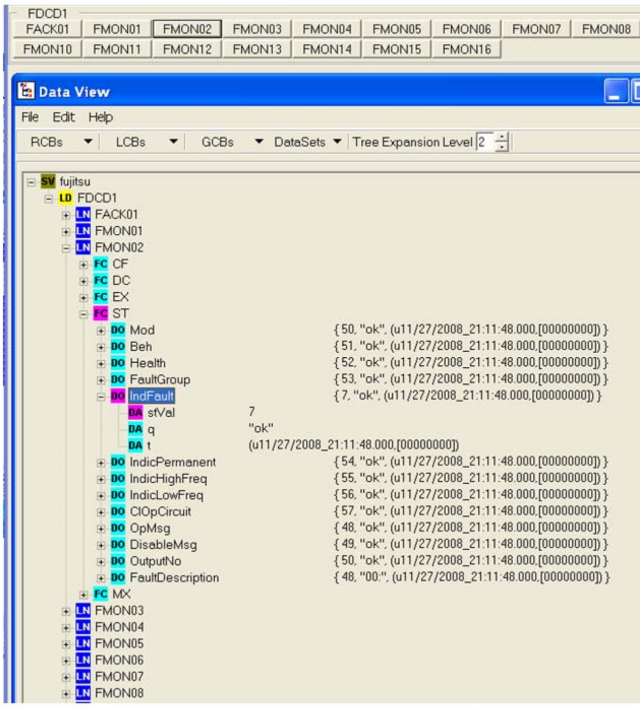


Fig. 10 Schematic representation of the communication for a fault signal.

D. *Testing the server*

For testing the server a software tool IEDScout (produced by the company Omicron electronics) was used. The IEDScout provides client functionality for developers of IEC61850 IED- servers. The realization of the IEC-Server Protocol Stack, so that it can be IEC 61850 conform, implies respecting the protocol imposed by the norm [1]. For this purpose, the IEDScout represents the ideal tool to verify the structure and the functionality of the device under development.

The IEDScout shows the structure of the logical nodes together with their values content in the fault data collection device as it can be seen in Fig 11. This signifies a successful initialization and corresponds to a server which is IEC 61850 conform.



```
SV fujitsu
  LD FDCD1
    LN FACK01
    LN FMON01
    LN FMON02
      FC CF
      FC DC
      FC EX
      FC ST
        DO Mod                        { 50, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO Beh                        { 51, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO Health                     { 52, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO FaultGroup                 { 53, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO IndFault                   { 7, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
          DA stVal         7
          DA q             "ok"
          DA t             (u11/27/2008_21:11:48.000,[00000000])
        DO IndicPermanent             { 54, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO IndicHighFreq              { 55, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO IndicLowFreq               { 56, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO ClOpCircuit                { 57, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO OpMsg                      { 48, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO DisableMsg                 { 49, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO OutputNo                   { 50, "ok", (u11/27/2008_21:11:48.000,[00000000]) }
        DO FaultDescription           { 48, "00:", (u11/27/2008_21:11:48.000,[00000000]) }
      FC MX
    LN FMON03
    LN FMON04
    LN FMON05
    LN FMON06
    LN FMON07
    LN FMON08
```

Fig. 11. The structure of our logical nodes shown by the client (IEDScout).

VI. CONCLUSIONS

Gaining acceptance worldwide, IEC 61850 is the first and only global standard that considers all the communication needs within substations. The standard has no models for signaling faults. Therefore an IEC 61850 conformant fault data collection device is here introduced. Two new specific logical nodes were defined by using attribute types of the standard. These additional nodes were implemented in a self made server which was tested through a connection with an hypothetic client. The client is represented by a software tool named IEDScout which provides client functionality for developers of IEC61850 IEDs servers. The fact that the IEDScout is able to represent the structure and the functionality of the device under development reveals a server which is IEC 61850 conform.

VII. REFERENCES

[1] *IEC 61850 Communication networks and systems in substations*, International standard, Part 1 to 10, 2003.

[2] S. Olson, "IEC 61850: Where are We Headed with Substation Communications?", *Utility Automation&Engineering T&D*, 2005.

[3] K.H. Schwarz - An introduction to IEC 61850. Basics and user-oriented project-examples for the IEC 61850 series for substation automation. *Vogel Verlag*, Würzburg, 2005.

[4] K.P. Brand - The Standard IEC 61850 as Prerequisite for Intelligent Applications in substations, Power Engineering Society*, IEEE General meeting* 6-10 June, p. 714-718, 2004.

[5] *DIN 19235: Measurement and control; signalling of operating conditions, German standard,* 01-Mar-1985.

[6] C. Brunner – IEC 61850 for Power System communication, *Transmission and Distribution Conference and Exposition, T&D IEEE/PES*, 21-24 April, 2008, pp 1- 6.

VIII. BIOGRAPHIES

Sidonia Mesentean is born in Musatesti (Romania), on March 11th, 1976. She graduated the Avram-Iancu high school, and she obtained her degree in chemistry informatics at Babes-Bolyai University in Cluj-Napoca. In 2001 she went on to obtain a Ph.D. in bio-informatics at Heidelberg University.

Recognizing that a revolution was beginning in the energy world, she changed fields and started to work at a project proposed by the Reinhold Würth University of the Heilbronn University in Künzelsau. During this project she started to get experience in the field of IEC 61850.

Heinz Frank received the diploma degree in electrical engineering from the University of Stuttgart in 1979 and the Ph.D. from the same university in 1985. Until 1991 he was at a machine tool company in charge for the development of computer control systems for manufacturing systems. In 1991 he became a professor for automation engineering at the Reinhold-Würth-University of the Heilbronn University in Künzelsau. His current projects have their focus on industrial communication systems and on fast mechatronic systems.

Klaus Fleischmann is born in Backnang (southern Germany) on October 5th 1964. After technical high school and military service, he studied microelectronics at the University of Applied Sciences in Esslingen and graduated as an engineer.

After his studies, he spent one year as an internee in the U.S.A, performing the commissioning of press lines at Schuler Inc. in Columbus, Ohio.

Since January 1, 1994 he has been chief development officer at UNITRO-Fleischmann in Backnang. His focuses are: fault and alarm systems for power generation and industry, and electric components for building and industrial automation.

Markus Stuhler is born in Gaildorf (southern Germany) on August 7th 1972. After junior high school, he apprenticed as a skilled worker at ANT Nachrichtentechnik GmbH in Backnang and received his certificate as an industrial electronics and equipment technician.

After completing a period of community service and 2 years in further training at the College for Technology in Stuttgart, he graduated with a diploma as state recognised technician in electronics and data technology.

Since 1998 he worked as hardware and software developer at UNITRO-Fleischmann in Backnang. His focuses are: development of fault and alarm systems, and electronic components for building and industrial automation based on Industrial Ethernet and LON technology.