Proceedings of the IASTED International Conference
POWER AND ENERGY SYSTEMS
May 13-15, 2002, Marina del Rey, California, USA

USING UCA STANDARD TO MODEL HYDROELECTRIC POWER PLANT AUTOMATION SYSTEM

Msc: Carlos A. Villacorta Cardoso
Cavica@usp.br

Prof Dr : Jose Antonio Jardini
Jardini@pea.usp.br

Prof Dr : Luiz Carlos Magrini
Magrini@pea.usp.br

Department of Electrical Engineering - University of São Paulo
Address: Av Prof. Luciano Gualberto No 380, São Paulo, SP-Brazil
Brazil

ABSTRACT

Integration achievement the is one of the main problems in the automation systems. At present, there is a universe of equipment with different protocols that need to be intercommunicated, something that always happens at all Automation hierarchical levels. The deregulation of the utilities have however, originated another needs such as: integration, consolidation and dissemination of information, in real time, in a fast and precise way within utilities, and among them too. Having these issues considered, the EPRI (Electrical Power Research Institute) has proposed the UCA (Utility Communication Architecture) standards, in order to try to solve the interoperability problems.
The extension of those standards in the modeling of objects and functions of power plants, originally proposed for the substation Automation systems, have been researched by the authors. This work presents a methodology to model and implement the UCA standards in Power plant automation. The case study corresponds to a hydropower plant start up function. The modeling is done based on an existing automation system.

KEY WORDS: Generation, UCA, GOMSFE and CASM.

1. Introduction

Nowadays, most of the automation systems have a strong dependence on both; information and communication systems. Furthermore, there are a lot of vendors who develop products with different protocols and data structures. As a result, this process has created "islands of information"; thus, these systems show many difficulties with the integration of such islands. Therefore, communication among them becomes complex, costly and sometimes impossible due to the lack of available specifications.
To cope with such situation EPRI (Electrical Power Research Institute) has proposed the UCA (Utility Communication Architecture) standards. This is a standard-based approach to utility communication, which provides a wide scale integration at reduced cost and solves many of the most pressing communication problems for the current utilities. The UCA is designed to be applied throughout the functional areas within the electric, gas and water utilities [1], [2].
The UCA includes detailed object models specifying the format, representation and the meaning of the utility data. This modeling effort goes far beyond the scope of any other utility communication approach, and provides an unprecedented level of multivendor interoperability.
This work shows how the UCA is used to model the hydropower plant automation system. It indicates the way the information of the automation system is organized in compliance with the UCA objects and the UCA models behavior.
The automation system data about the hydropower plant were obtained by a survey for some Brazilians utilities.

2. UCA

The UCA first version consisted of a set of protocol specifications with two models: firstly, a seven-layer set in compliance with the OSI model and secondly a three-layer set oriented to attend the real-time requirements. The common protocol scheme of UCA provides significant additional benefits for the utilities to cope with an increased communication requirements due to the deregulation [3].
The main barrier to adopt the UCA first version standard was the lack of detailed specification on how the protocols would actually be used in the utility field device. The rich functionality and broad generality of the protocol application layer MMS (Manufacturing Message Specification), particularly meant that without further specification, the field devices could be implemented in the utility applications using a variety of services and procedures that resulted in a constant lack of interoperability [4]. In the Version 2.0, there are a number of efforts which were initiated to develop detailed object models of common field devices, including definitions of their associated algorithms and visible communication behavior through the communication system. The results of these efforts are stated in two documents: the Generic Object Models for Substation and Feeder Equipment GOMSFE) and the Common Application Service Models (CASM).

Table 1. Common Class "AI" Analog Input.

Common Class: AI *Analog Input*		
Name(Common Componentes)	**Data Type**	**m/o**
i	INT16S	m
f	FLT32	O
q	BSTR16	O
t	BTIME6	O

Table 2. Brick: GAIN "Generic Analog Input".

Brick: *GAIN*			
FC	**Name**	**Common Class**	**Description**
MX	In<n>	AI	Generic Analog Input
CF	GAIN.MX	ACF	Configuration for GAIN.MX
DC	GAIN.MX	D	Description for GAIN.MX
RP	brcbMX	BasRCB	Controls reporting of Measurements

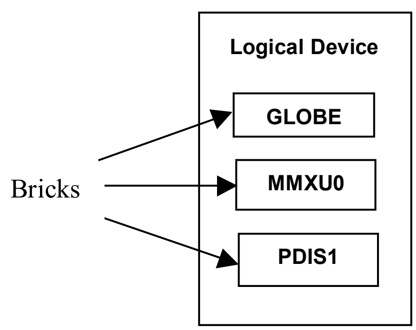

Figure 1. The bricks are aggregated to form a Logical Device.

2.1 GOMSFE

This document specifies, through the technique oriented object, the elements used in the modeling of the automation objects and their hierarchy. These elements are: types of data, common components, common classes, bricks, Logical Devices.

The **standard types** of data determine the format, number of bits to communicate the value, and the range of possible values (e.g. an unsigned eight-bit integer is defined as INT8U).

Based on the previous standard data types of, the **common Components** are also defined. They represent elementary components used in the definition of the object-class. The names of the common components are constructed from abbreviations that reflect the meaning of the variable. An example of common component is "b", which represents a binary value with a standard type of data BOOL (logical type that can be "true" or "false").

The **Common classes** are groups or structures of common components which are the attributes of the modeled objects. The common classes represent the classes more frequently used. "AI" (showed in Table 1) is an example of a common class used to define an analog input.

Bricks are the blocks or basic blocks of construction. The bricks are a collection of objects. These objects can be composed by common components and structures based on Common Classes, (Table 2). Bricks can be defined also as groups of associated objects addressed to be used or re-used in a private function. For the present work effects a brick is basically the representation of a field instrument.

A brick provides the standard interface definition for the outside world to communicate with field device controllers. Figure 1 show how a group specialized of devices (bricks) that are modeled by **Logical Devices** (L.D.). Moreover, the L.D. models are intended to be guidelines to commonly agreed functions and applications. Likewise, the devices are modeled using the bricks, which are groups of data objects. The **Data Objects** are the elemental information unit in UCA. It can be a single data or set of data [5].

2.2 CASM

The UCA Common Application Service Model (CASM) provides a common set of communication function for data access, reporting logging, control applications and related support. The use of these services allow for the:

- separation of the models from service and communication details
- high level of application interoperability
- reduced integration and development costs through the use of common mechanisms for data access and communication establishment [1], [6].

The Common Application Service Models are defined using object modeling techniques. For this reason, the Field device models can incorporate these services by specifying which objects within their models inherit the object classes defined within this document. For example, if a model of a utility field device contains a control object, which requires a two-step commit (select-before-operate), the object should inherit the attributes and methods associated to the corresponding object class. Besides, this document specifies the mapping of CASM in the layer application protocol MMS.

One of the UCA applications is the project conducted in conjunction with the Pacific Gas and Electric (PG&E) and Houston Light and Power (HL&P), this experience resulted in UCA version 1.0 [1]. Another project that used the UCA standard was the United Power Association project (UPA) for distribution automation. The goal of this project was to show the use of the UCA compliant hardware and software is an effective mean to achieve access in real time data for multiplying users. This project was sponsored by UPA, the National Rural Electric Cooperative Association (NRECA) and EPRI. The system consists of three control centers, two distribution substations and several pole-top devices. This paper also mentions another project being this the City Public Services of San Antonio (CPS of San Antonio) using the draft 2.0 [2].

General Electric has an experience with a good combination: UCA version 2.0 and Ethernet technology. The G. E. paper describes how to fulfil the primary functional requirements in hardware (i.e. scalability, reliability and performance), which demonstrated it is shown that it is possible using the Ethernet network. With respect to software requirements such a work intended to use the "off the shelf" standards, which were achieved [3]-[7]-[8].

This paper presents the use of the UCA standards to model a Hydro Power Plant applications; however it is yet an ongoing work in the Power and Automation laboratory of the University of São Paulo.

3. Advantages of Using UCA

The difference between UCA and most of the previous utility protocols is in the use of device object models and their components. As seen in the last section, these models define common data formats, identifiers and controls for the substations and feeder device, namely: measurement unit, switches, voltage regulators and relay models. Besides, these models specify standardized behavior for the most common device functions. Therefore, the standardization of the data representation and behavior of the automation objects allow for the multivendor interoperability, thus, improving the integration [1].

A natural benefit of using object oriented technique to model automation systems is the ability to "self define" their data objects and the object commonality among the manufacturers. The latter result in a tremendous time saving to add new variables without the need to change the memory map of the device.

With these briefly described experiences, it is possible to conclude that the standardization is indispensable to promote both the interoperability and the integration of the electric system control. The UCA is an emerging standard and its study and application will be very useful in the future.

4 Methodology

4.1 Identification of Logical Devices

This work, relies on the advantage of making the correspondence between Subsystems and Logical Devices because a subsystem consists of a group of instruments mainly in charge of a specific process control. Therefore, each Logical Device will have the particularities of every process in the subsystem. The automation subsystems were identified (Table 3).

In our model the GOMSFE Logical Device corresponds to the above mentioned subsystems. In the modeling process the data which belongs to every subsystem was identified, too.

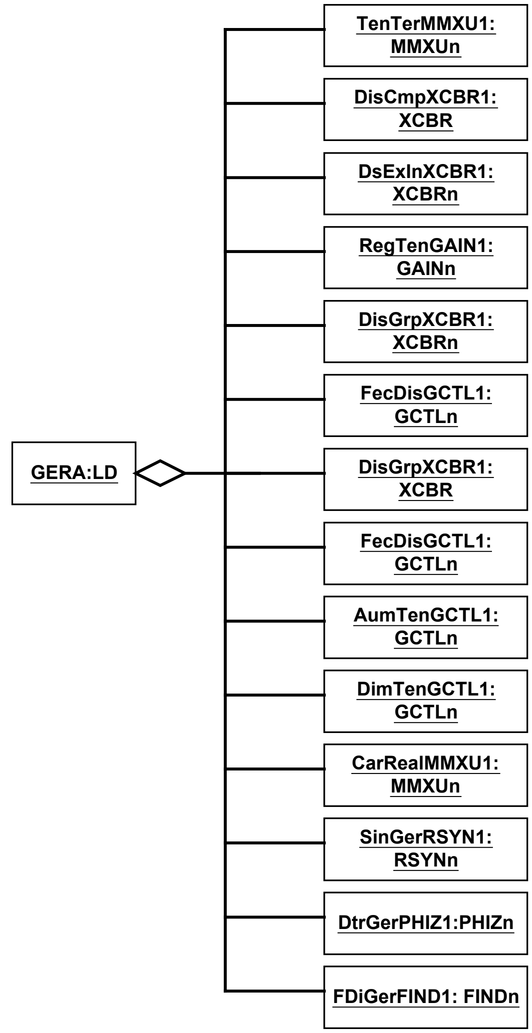

Figure 2. Logical Device Modeling: GERA in a UML diagram class

4.2 Modeling the Logical Devices

The model of every Logical Device is obtained from the bricks' aggregating that corresponds to every devices belonging to the subsystem. One example of Logical device modeling seen in figure 2, where the GERA Logical device is represented using the UML (Unified Model Language). GERA corresponds to the machine Generator's instruments. All the bricks are aggregated to form the LD and each is an instance of the generic standard bricks; e. g., TenTerMMXU1 is an instance of the MMXU standard brick and is aggregated to form the LD, GERA.

4.3 Configuration of Generic Bricks

As an example of the bricks modeling the TenTerMMXU1, first instrument of the Table 4, can be used. This corresponds to the "terminal voltage meter".

Table 3. Subsystem Instrumentation at the Hydro power System start up

Subsystems	Abbreviation
Generator	GERA
Distributor	Dist
Braking System	FrAr
Cooling System of the shaft Generator bearing	RMGG
Cooling System of Generator machine	SRG
Shaft Sealing System	SVE
Sealing System for shaft Maintenance	SVME
Strut bearing Cooling System	RME
High pressure oil system of the Strut bearing	OAP
Turbine Guide bearing Cooling System	RMGT
Generator Guide bearing Supervisor System	SMGG
bearing Strut Supervisor System	SME
Turbine Guide bearing Supervisor System	SMGT
Pentstook	SCF
Step up Transformer	TE
Command Unit	COM
Protection Unit	PROT

Table 4. Example of a Logical Device (Generator).

LD: Gera Generator Devices	GOMSFE Standard brick	GOMSFE Nomenclature
Terminal voltage	MMXU	TenTerMMXU1
Field Breaker	XCBR	DisCmpXCBR1
Primary excitation Breaker	XCBR	DsExInXCBR1
Voltage Regulator	GSPT	RegTenGSPT1
Breaker of group	XCBR	DisGrpXCBR1
Command close breaker	GCTL	FecDisGCTL1
Command increase the voltage	GCTL	AumTenGCTL1
Command reduce the voltage	GCTL	DimTenGCTL1
Reactive load	MMXU	CarReaMMXU1
Synchronism	RSYN	SinGerRSYN1
Differential Relay	PDIF	PdiGerPDIF1
Ground detector	PHIZ	DTrGerPHIZ1
Circuit Breaker fault	FIND	FDiGerFIND1
Circuit Breaker fault Relay	PBRO	FDiGerPBRO

The structure of the TenTerMMXU1 model can be observed in Table 5 and its generic model in figure 3. It only use the "V" class that is also an instance of the common "WYE" Class, which indicate that the measurement is performed in three phases. The common class ACF is considered because it is necessary to set up the variable properties of the “V” class. Additionally, the common class “BrcbMX” it is necessary because it defines the behavior of the “TenTerMMXU1” measurements.

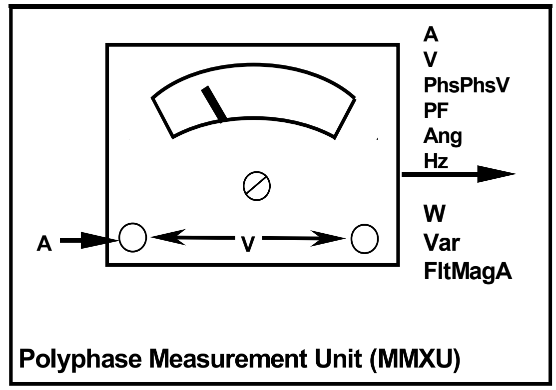

Figure 3. Generic Brick MMXU used to model TenTerMMXU1.

Table 5. The TenTerMMU1 brick

FC	Object Name	Class common	Description
MX	V	WYE	Voltage on phase A, B, C to Ground
CF	All MMXU.MX	ACF	Configuration of ALL included MMXU.MX
RP	BrcbMX	BasRCB	Controls reporting of Measurements
AS	LogDev<n>	TBD	Defines path for Peer to Peer Communication

4.4 Distributed Logic Implementation

After modeling all Logical devices, bricks and its Objects, the following step is to model the control logic, therefore In this study case (the Hydro generator Start up function), the logic diagrams should be passed to the UML sequence diagrams, where diagrams the communication should be modeled by using the CASM standard.

Figure 4 shows, an example of the variables’ check ups. This logic function is performed inside two Logical Devices: COM and GERA. These logical devices have to get the states of the following object conditions:

- If the electronic Opening adjusting reference limiter is at start up position. This state is obtained from the LD: Dist (Dist.LmEleGSPT1);
- If the Opening mechanical limiter is at 100%. This state is obtained from LD: Dist (Dist.LmMecGAIN1).
- If the main regulation oil pump is at set up pressure. This state is obtained from LD: Dist (Dist.BPrORPUMP1).
- If the cooling water lubrication oil is at normal flow. This state is obtained from LD RME (RME.WFluGAIN1);
- If the generator cooling water is at normal flow. This state is obtained from LD: RG (RG.FSaWGAIN1);
- If the lubrification oil blocked valve is open. This state is obtained from RME (RME.ValBlocGCTL1).
- If the injection oil system is working propertly. This state is obtained from LD OAP (OAP.OlePresGAIN1);

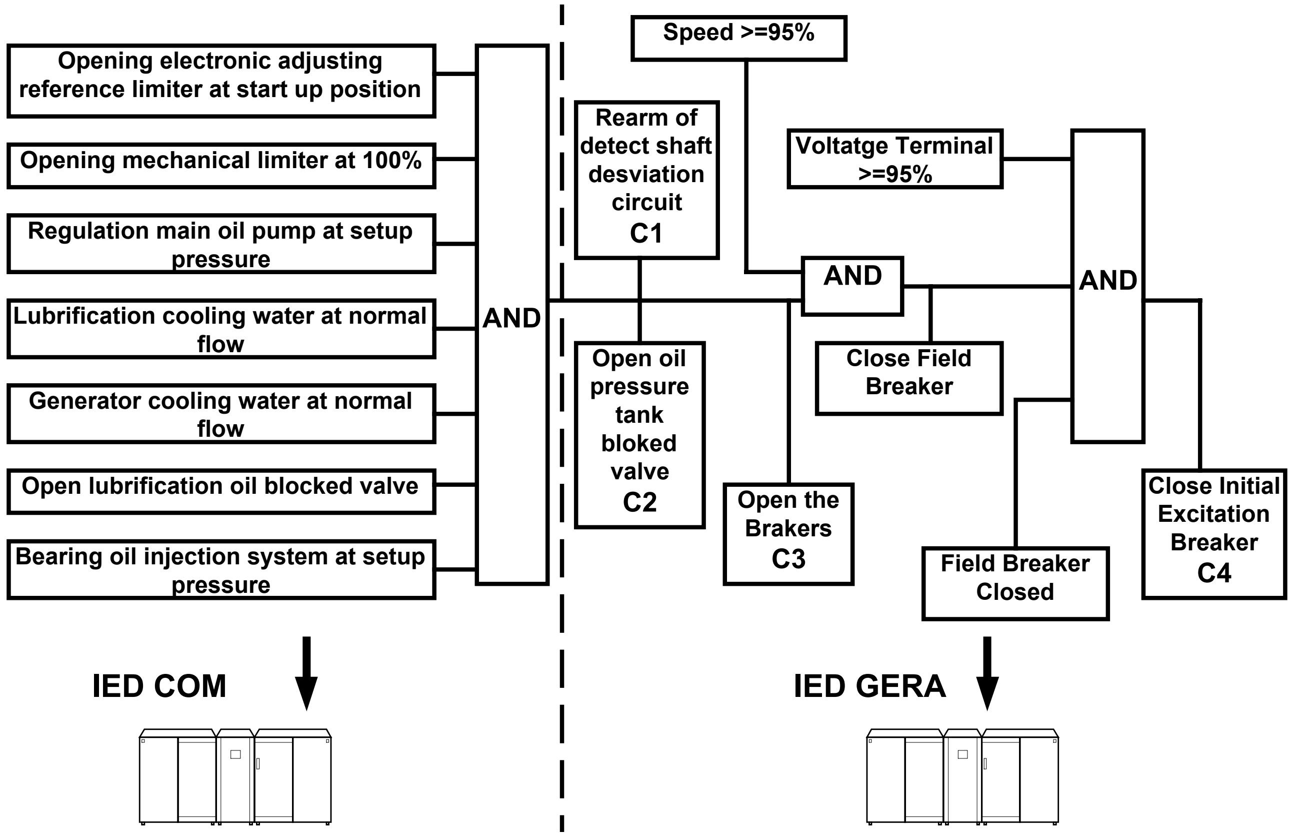

Figure 4. Part of the Hydropower Plant logic start up

- If the speed is at 95%. This state is obtained from LD: Dist (Dist.RotaGAIN1.MX);
- If the terminal Voltage measurement is at 90% of the rated voltage. This state is obtained from the L.D. GERA (GERA.TenTerMMXU1);
- If the Field Breaker is closed. This state is obtained from LD: GERA (GERA.DisCmpXCBR1);

Depending on the above results the LD: COM should send three commands:

- Rearm the shaft deviation detector circuit. This command is sent to LD: GERA
- Open the oil pressure tank blocked valve. This command is sent to LD: OAP
- Open the braker. This command is sent to LD:FrAr.

Similarly, the LD: GERA should send two commands:

- Close Field Breaker;
- Close initial excitation breaker

Besides, those external objects belonging to another Logical Devices should be check up. One example, is when the Distributor has to check up its internal variables as is the case of the "rotation Speed".

The Communication between Logical Devices is done following the standardized communication models standardized by the CASM.

Figure 5, presents the interactions used to get the logic shown in figure 4. The logic gates are implemented in each LD.

Figure 5, shows the event sequence and the CASM communication services used to implement part of the logic used as an example (Figure 4). However, it was necessary to set up the Objects' behavior previously. In this case two models Reporting Service and Device Control are used, which are defined in the CASM. Basically the bricks: Dist.LmEleGSPT1 (Electronic Limiter); Dist.LmMecGAIN1 (Mechanical Limiter); RME.WFluGAIN1 (Lubrication oil cooling water); Dist.BPrORPUMP1 (The regulation main oil pump); RG.FSaWGAIN1 (The generator cooling water); and OAP.OlePresGAIN1 (Injection oil System), have to be configured to report their variable states, periodically. It should be done setting each of the Report Control Block (RCB) existing in every brick.

An example of RCB set up can be observed when the LD: COM demanded the water flow measurement (RG.FSaWGAIN1.MX.Flw). This case applies the SetDataObjectValue service, with the need of setting up the RG.FSaWGAIN1.RP.brcbMX. The same procedure has to be applied for the following variables: Dist.LmEleGSPT1,Dist.LmMecGAIN1,RME.WFluGAIN1, Dist.BPrORPUMP1. The L.D. COM will be able to receive those variables in the future, with no further requests.

The commands previously referred to in figure 4, (C1, C2, C3 and C4) are examples of Select Before Operating-SBO, (not thoroughly shown in figure 5). The SBO Control capability allows for the control modeling which requires a two-step commit procedure. A DataObject (variable), which contains an SBO Control Object may not be written by a client unless the enclosed SBO Control Object is in the SELECTED state for that Client. The state of the SBO Control Object is set to SELECTED through the use of the Select Operation. Several conditions may cause the SBO Control Object to be reset to the DESELECTED state, depending on the style the SBO is employed. In this part, the SBO has been used to model the Breakers' operation such as the Close Field Breaker and Close initial excitation breaker. An example of SBO operation is shown in figure 6. The sequence of messages observed permits seeing how firstly, the client LD:COM selects the "Braker" in the LD:FrAr and after the confirmation, the LD:COM by using the service SetDataObjectValues operates to trip the Brakers.

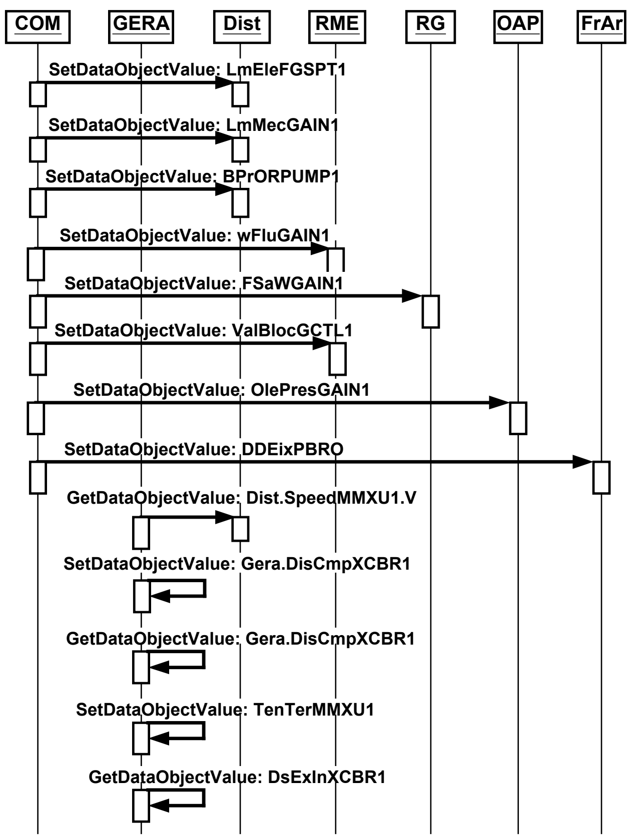

Figure 5. Part of the distributed logic to start up the Hydropower Plant expressed in UML dynamic diagram using the CASM services.

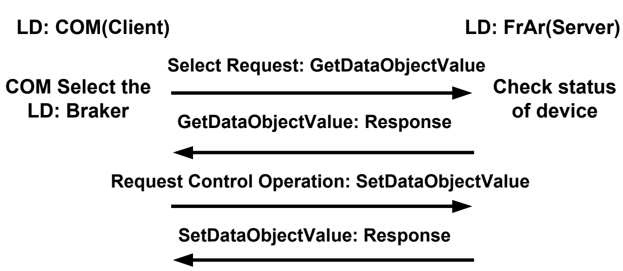

Figure 6. Interaction between client and Server in a SBO Command.

5 CONCLUSION.

This work shows the UCA facility to model an Automation system in a Hydropower Plant. The existing instruments in the system can be modeled by using the GOMSFE standard. Owing to the GOMSFE object models the implementation of a distributed logic from the logic diagram is possible. The interaction between these objects can be mapped following the CASM standard.

It is visibly interesting to project an IED (Intelligent Electronic Device) network (like LAN), which can implement this distributed logic. Each IED would correspond to a UCA Logical Device model system. The next step, still being carried out, is researching the object automation and researching communication services' mapping, using open technologies to permit a total integration between control networks and management networks, such as CORBA "Common Object Request Broker Architecture".

REFERENCES

[1] K. Schwarz, IEEE UCATM AND IEC 61850 APPLIED IN DIGITAL SUBESTATIONS. Available: http://www.NettedAutomation.com. NettedAutomation GmbH, 2000.

[2] R.F. Gilbert, J.L. Goodin, K. Nelson, The benefits of a distribution automation system using the utility communications architecture (UCA), *Proc. 42nd Annual Conference on Rural Electric Power Conference*, St. Louis, MO, USA, 1998, c3-1-5.

[3] M. Adamiak and W. Premerlani, Data communications in a deregulated environment. *IEEE Computer Applications in Power*, Volume: 12, Issue 3, July 1999, 36 –39.

[4] Electrical Power Research Institute, EPRI, *Introduction to UCA version 2.0 - draft 1.0* (EPRI, 1997).

[5] Electrical Power Research Institute, EPRI, *Generic Object Models for Substation & Feeder Equipment- draft 0.92 (GOMSFE)* (Palo Alto, EPRI, May, 2001).

[6] Electrical Power Research Institute, EPRI, *Common Application Service Models (CASM) and Mapping to MMS - draft 1.6* (EPRI, 1998).

[7] M. Adamiak and D. Baigent, Practical Considerations in Application of UCA GOOSE. Proc, *"Georgia Tech Relay Conference"*. May 2000a.

[8] M. Adamiak and D. Baigent, "UCA 2.0 The GE Experience". Available: http://www.scc-online.de/download/UCA/UCA-Articles/. 2000b.