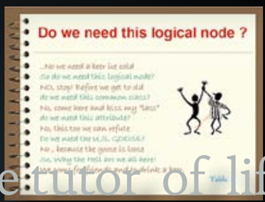
# 61850 Award

## WG10, WG11, WG12


# Do We Need this Logical Node?

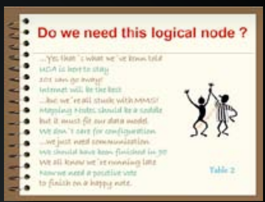
History is the tutor of life

### Table 1



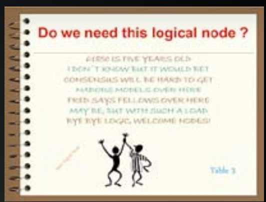

..No we need a beer ice cold
So do we need this logical node?
No, stop! Before we get to old
do we need thid common class?
No, come here and kis my "Lass"
do we need this attribute?
No, this too we can refute
Do we need the U.S. GOOSE?
No, because the goose is loose
So, why the Hell are we all here?
We come for friends and to drink a beer.

### Table 2




..Yes that's what we've been told
UCA is here to stay
101 can go away!
Internet will be the best
...but we're all stuck with MMS!
Mapping Nodes should be a soddle
but it must fit our data model!
We don't care for configuration
...we just need communication
We should have been finished in 98
We all know we're running late
Now we need a positive vote
to finish on a happy note.

### Table 3

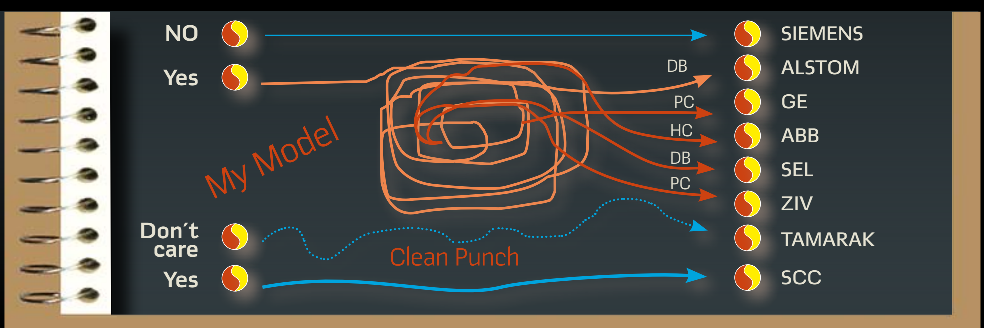



61850 is five years old
I don't know but I would bet
consensus will be hard to get
Madors models over here
Fred says fellow over here
May be, but with such a load
Bye Bye logic, welcome nodes!

### Table 4

DB: Dimpled Ballot

PC = Pregnant Chad
HC = Hanging Chad




The rhyme created at the WG 10, 11, 12 meeting in Valencia (Spain) in 2000.

by Walter Schossig, Germany, and Thomas Schossig,
OMICRON electronics GmbH, Austria



# Protection History

## The GOOSE departs

**In the last issue the way to IEC 61850 was described.** We explained the activities of UCA and EPRI, covered MMS. Ethernet entered the substation and first interoperability events took place. The discussion was not only about the protocol to be used, but also for the data modeling. Under auspices of the EPRI UCA the Generic Object Models for Substation and Feeder Equipment (GOMSFE) have been developed. The goal of GOMSFE was to facilitate vendors in implementing the UCA substation and feeder elements of the power system into a product design. It did not, however, describe the mapping of the objects onto the application layer. This mapping was described in the UCA 2.0 Common Application Services Model and Mapping to MMS (CASM). The building blocks, or "bricks" modeled in GOMSFE are aggregated to form the logical device models that are contained by the remote server and represent real world devices.

The modeling process is based on the definition of a library of basic, common objects and standardized, reusable, bricks (basic building blocks built from common objects using standard data types, common data classes, and functional component groupings) to add further levels of functional detail and specification. The individual objects and object components are defined using standard types and the objects and models are defined at the most simplified and common classification level to allow the construction of widely varying models. These objects are given machine readable names using standard abbreviations. Vendor extensions to the models may be made following the required GOMSFE model. The GOMSFE objects are mapped onto generic services of the application layer using the UCA Common Applications Services Model (CASM). The CASM provides a generic application layer service platform for communications throughout the utility architecture. The CASM is mapped onto the Application Layer protocol services specified by UCA for the particular application. UCA defines a set of communication profiles (protocol stacks) to support the application layer. The "building blocks" mentioned are that what we call "logical nodes" in IED 61850 today.

Although the goal in choosing the various profile layers for the substation implementation of UCA was to use "existing protocols", some enhancements were needed to meet the required functionality. One of these functions, in particular, was high speed device to multi-device communications. The MMS information report service was used to achieve this functionality and it was used to deliver a binary object model known as the Generic Object Oriented Substation Event or GOOSE. The idea of transmitting substation events quite quickly is as old as substation systems exist. And when ASEA introduced a high speed (4 ms) auxiliary tripping relay – the RXMS 1 in the mid-1970s this was new quality. And so the EPRI Report "RP 3599 Substation Integrated Protection, Control, and Data Acquisition – Requirements Specification" – published on Oct 31st, 1995 requires 4 ms for some applications. The diagram defining the 4 ms as "application to application", not just time on the wire, came later.

The first simulations at Fraunhofer Institute (supervised by Karlheinz Schwarz) took place in the 3rd quarter of 1996. The already mentioned Generic Object Models for Substation & Feeder Equipment (GOMSFE) Draft 0.4 was published in April 1997 and is from our knowledge the earliest use of the

**Walter Schossig** (VDE) was born in Arnsdorf (now Czech Republic) in 1941. He studied electrical engineering in Zittau (Germany), and joined a utility in the former Eastern Germany. After the German reunion the utility was renamed as TEAG, Thueringer Energie AG in Erfurt. There he received his Masters degree and worked as a protection engineer until his retirement. He was a member of many study groups and associations. He is an active member of the working group "Medium Voltage Relaying" at the German VDE. He is the author of several papers, guidelines and the book "Netzschutztechnik [Power System Protection]". He works on a chronicle about the history of electricity supply, with emphasis on protection and control.

**Thomas Schossig** (IEEE) received his masters degree in Electrical Engineering at the Technical University of Ilmenau (Germany) in 1998. He worked as a project engineer for control systems and as a team leader for protective relaying at VA TECH SAT in Germany from 1998 until 2005. In 2006 he joined OMICRON as a product manager for substation communication products. He is author of several papers and a member of standardization WGs.

term GOOSE. The GOOSE message was implemented as a connectionless ISO datagram. As a connectionless message, the Media Access Control (MAC) address of the sending device is known and subsequently loaded into the header of the GOOSE message. Also included in the header is the name of the sending device, time of the event that launched the GOOSE, and the expected time of the next GOOSE message. Data states are sent in pairs with (0,1) and (1,0) being the primary logical states, (0,0) being defined as a "transition" state, and (1,1) being undefined (see below for details) Each node receiving a multi-cast message needs to be able to determine from whom the message came and whether the data received was of interest to the receiving device. In establishing the procedure for how this would happen, GE worked with SISCO to develop the Self Mentoring And Re-Training or SMART-GOOSE. Self Mentoring is the process of finding out the address of the GOOSE sender to which a device would like to listen. During set-up, the engineer programs the receiving device with a list of devices from which it should expect to receive data. On start-up of the network, the Ethernet receiver in the device goes into "promiscuous" mode whereby all multi-cast messages are read and decoded. The name of the device sending the message is compared with the programmed list of "devices to listen to". If the names match, the receiving device stores the MAC address of the sending device in a high-speed hardware address comparator. Once all address / name matches have been made, promiscuous mode is turned off and all multi-cast messages are now captured based on a hardware address comparison.

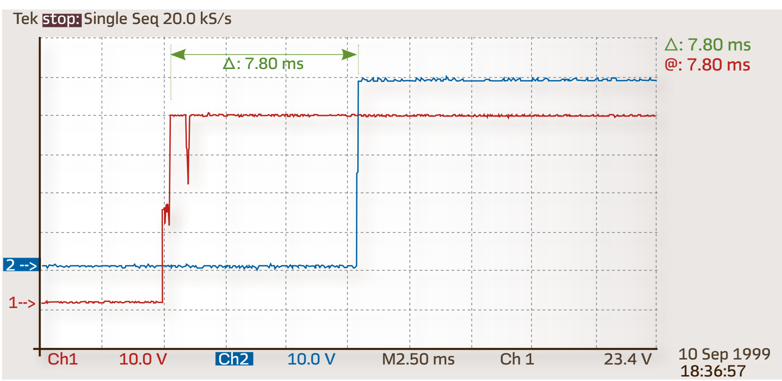
The "Re-Training" part of the SMART GOOSE comes into play when a relay is taken out of service or a CPU module is exchanged or upgraded. When the CPU card is changed, the corresponding MAC address for the Ethernet card changes as each Ethernet controller in the world has a unique 48 bit address. SMART GOOSE (on the receiving side) recognizes that the GOOSE message it was expecting is missing. In this scenario, the receiving relay goes back into promiscuous mode – again searching for a message with the name of a desired device inside. Once found again, the new MAC address for the new CPU / Ethernet controller is stored in the high-speed look-up table and the receiving relay once again turns off promiscuous mode and returns to normal operation.

One particular performance challenge was with GOOSE messaging. In the original implementation, GOOSE messages were processed in the same thread with all other

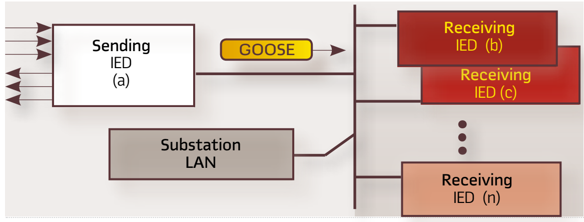
MMS data requests. As such, GOOSE message delivery times were averaging 15ms - somewhat beyond the goal of 4ms established by the UCA specification team. In revisiting the GOOSE implementation, a refinement was made to allow the GOOSE decoding software to operate as a separate thread in the processor. Making this change allowed received GOOSE messages to be processed separately from all other MMS messages and consequently, allowed GOOSE messages to be processed at a higher processing priority in the CPU. Additionally, the encoding and decoding of GOOSE messages was optimized savings additional milliseconds of

GOOSE processing time. As a result, GE was able to increase the speed of their GOOSE message 3 fold. Complete digital input to digital output timing includes digital debounce

## 1 Digital Input to Digital Output GOOSE Timing, GE, 1999




## 2 Peer-to-peer GOOSE communications




## table 1 GOOSE message DNA

| Bit # | Bit Pair | Bit Order / Value / Definition | 00 / 0 / State | 01 / 1 / State | 10 / 2 / State | 11 / 3 / State |
|---|---|---|---|---|---|---|
| 0, 1 | 1 | OperDev | Default Value | Preset | Pass Through | Invalid |
| 2, 3 | 2 | Lpck Out | Default Value | Preset | Pass Through | Invalid |
| 4, 5 | 3 | Initiate Reclosing | Default Value | Preset | Pass Through | Invalid |
| 6, 7 | 4 | Block Reclosing | Default Value | Preset | Pass Through | Invalid |
| 8, 9 | 5 | Breaker Failure Initiate | Default Value | Preset | Pass Through | Invalid |
| 10, 11 | 6 | Send Transfer Trip | Default Value | Preset | Pass Through | Invalid |
| 12, 13 | 7 | Receive Transfer Trip | Default Value | Preset | Pass Through | Invalid |
| 14, 15 | 8 | Send Perm | Default Value | Preset | Pass Through | Invalid |
| 16, 17 | 9 | Receive Perm | Default Value | Preset | Pass Through | Invalid |
| 18, 19 | 10 | Stop Perm | Default Value | Preset | Pass Through | Invalid |
| 20, 21 | 11 | Send Block | Default Value | Preset | Pass Through | Invalid |
| 22, 23 | 12 | Receive Block | Default Value | Preset | Pass Through | Invalid |
| 24, 25 | 13 | Stop Block | Default Value | Preset | Pass Through | Invalid |
| 26, 27 | 14 | BkrDS | Default Value | Preset | Pass Through | Invalid |
| 28, 29 | 15 | BkrPhsADS | Default Value | Preset | Pass Through | Invalid |
| 30, 31 | 16 | BkrPhsBDS | Default Value | Preset | Pass Through | Invalid |
| 32, 33 | 17 | BkrPhsCDS | Default Value | Preset | Pass Through | Invalid |
| 34, 35 | 18 | DiscSWDS | Default Value | Preset | Pass Through | Invalid |
| 36, 37 | 19 | Interlock DS | Default Value | Preset | Pass Through | Invalid |
| 38,39 | 20 | LineEndOpen | Default Value | Preset | Pass Through | Invalid |
| 40, 41 | 21 | Mode | Default Value | Preset | Pass Through | Invalid |
| 42, 43 | 22 | Event | Default Value | Preset | Pass Through | Invalid |
| 44, 45 | 23 | Fault Present | Default Value | Preset | Pass Through | Invalid |
| 46, 47 | 24 | Sustained Arc | Default Value | Preset | Pass Through | Invalid |
| 48, 49 | 25 | Downed Conductor | Default Value | Preset | Pass Through | Invalid |
| 50, 51 | 26 | Sync Closing | Default Value | Preset | Pass Through | Invalid |
| 52, 53 | 27 | Reserved | Default Value | Preset | Pass Through | Invalid |
| 54, 55 | 28 | Reserved | Default Value | Preset | Pass Through | Invalid |
| 56, 57 | 29 | Reserved | Default Value | Preset | Pass Through | Invalid |
| 58, 59 | 30 | Reserved | Default Value | Preset | Pass Through | Invalid |
| 60, 61 | 31 | Reserved | Default Value | Preset | Pass Through | Invalid |
| 62, 63 | 32 | Reserved | Default Value | Preset | Pass Through | Invalid |

by Walter Schossig, Germany, and Thomas Schossig,
OMICRON electronics GmbH, Austria



time on the input and contact actuation time on the output. Figure 1 shows a GE- relay input to relay output timing diagram.

The second earliest use of the term GOOSE we found was in a paper by JTT presented in November 1997. The title was "LAN Congestion Scenario" and already mentions the "MMS GOOSE". Cites results from ComNet simulation done earlier that year. Paper also says "All messages use an unacknowledged protocol." At the WG 10-12 Scenarios Task Force Meeting in February 1998 JTT presented the above paper and was asked to prepare a summary paper covering the ComEd LAN congestion scenario and ComNet III simulations.

The first draft of the reuquested summary for IEC was available in April 1998. Ultimately became Annex B 1998-27-10 to 57WG10, and also IEEE Paper 99WM686 "LAN Congestion Scenario and Performance Evaluation" by Simon, Sufana and Tengdin.

The GOOSE should work as a peer-to-peer communication (Figure 2.)

When they were conceiving GOOSE as part of UCA2, they knew that they had to make it as good or better in performance and reliability than the existing wired practice

**The goal of GOMSFE was to facilitate vendors in implementing the UCA substation and feeder elements of the power system into a product design.**

3 **GOOSE repetition mechanism**


4 **UCA demonstration with ABB, SIEMENS, OMICRON (2001)**

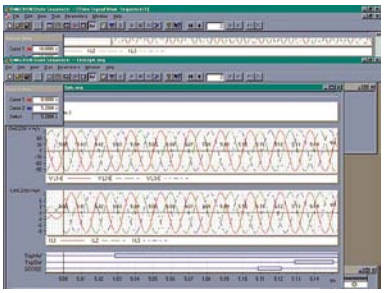

5 **UCA demonstration with GE, OMICRON (2001)**

6 **UCA demonstration with ALSTOM, OMICRON (2001)**


7 **UCA demonstration with SEL, OMICRON (2001)**

**Very soon the question for testing solutions for protection engineers occurred.**

**8 UCA demonstration measurement results (2001)**



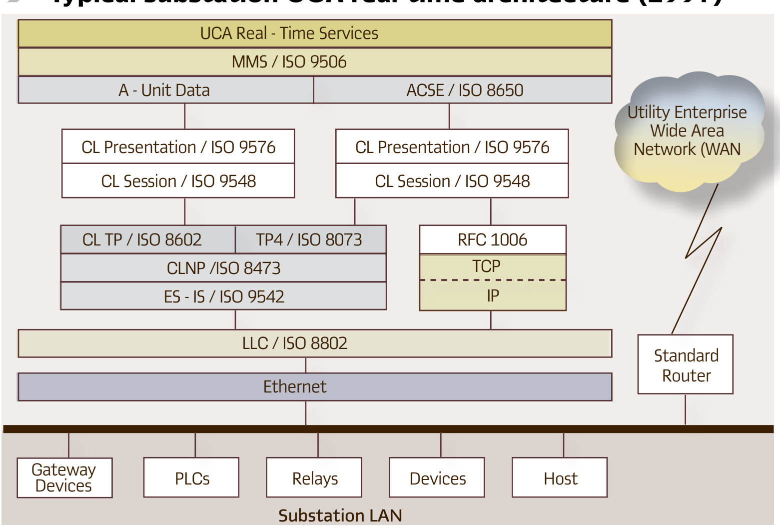
Lucy Goosy

and it needed to come at a lower cost too. While not a day one objective this is what they were aiming for. The requirements were derived from the protection practice of using a relay's contact output to drive the next device's opto-isolator input. There were lots of naysayers, but the engineers kept thinking that if protection and control engineers drove the requirements, without sacrificing our principals, they would get it right. And they clearly did.

With GOMSFE (Generic Object Models for Substation & Feeder Equipment) the model could be described.

As they discussed what could go wrong, Mark Simon (see his letter in September 2015 issue) came up with the repetition mechanism so that "normal traffic" would not overwhelm the 10 MB Hub's (circa 1995). No reason to send the "I'm ok" message every 4ms. The message counter in the GOOSE is available to detect missed messages so that a breaker failure scheme would not be delayed if the initial message were missed. They wanted any generator stability critical clearing times to be adhered to. From tests they found that more than 20 GOOSE devices on the same LAN segment would start to produce collisions that might delay message receipt. So, they knew that network design was essential in assuring that the time requirements were going to be met.

**9 Typical substation UCA real-time architecture (1997)**




As time went on Hubs were replaced with switches. While making the design easier, we must never forget that GOOSE needs a suitable network to make sure that the performance and redundancy requirements will be met under the most severe of power system fault conditions. How to be sure that they came up with something that they could configure in an actual relay? The experts in the room included folks that could assure us that their thoughts could be put into firmware and interoperate between vendors. Who was "they"? They called themselves the "Chicago 7". Chicago is the "birthplace of GOOSE". The "Chicago Seven" are Kay Clinard, George Schimmel, Herb Falk, John Tengdin, Mark Simon, Charlie Sufanna and Alex Apostolov. In the week of May 17, 1998 the "Chicago Seven + One" meeting at Marriott Courtyards, Woodale, IL (USA) with Jim Whatley via teleconference took place. On May 26, 1998 - this is the date on a Mark Simon Excel document- the GOOSE requirements have been published. This document defined the GOOSE DNA, and was the answer to the question – How do you tell one GOOSE from another? By its DNA!

Multiple point to point messages have been combined into a container called GOOSE DNA, thereby reducing the number of messages to be sent on the LAN during an event. An IED launches a GOOSE sequence for a change in state of any of its DNA elements (Table 1.)

Normal LAN communications utilizes acknowledged responses to guarantee proper delivery. In other words, the message is repeated until both sender and receiver are satisfied. This provides reliability without a concern for delivery time. Relaying requires reliability without sacrificing speed. Rather than wait for an acknowledgement, messages are repeatedly retransmitted. Intervals between retransmissions increase over time to minimize LAN congestion. Message intervals extend to a maximum time so the IEDs can be aware of each other's health. Also part of the message are GOOSE common components such as: time to live, repeat interval, back time, and sequence numbers. An IED continuously examines the GOOSE message variables such as time to live and hold time while making a judgment as to the validly of the GOOSE DNA. A GOOSE Variable Processor will use the information in the Default Memory to replace bad data, so that an IED will have predictable data if the received data from a particular IED is either invalid, missing or to old. The GOOSE Variable Processor would be responsible for notifying the IED processor that there is new data.

MASK Memory is provided for test purposes. It allows, via programming, the user to force any elements of any of the subscribed GOOSE DNA to a particular state. This allows the testing of individual protection elements, logic functions and the disabling of certain elements without having to change the IED's Boolean logic set. This is meant for short term testing. An alarm flag will be sent by the IED to indicate that the IED is in a test mode.

The details of the coding have been described in "Proposed IEC 61850 GOOSE ASN.1 Grammar" by George Schimmel and amended by Herb Falk in March 2001.

by Walter Schossig, Germany, and Thomas Schossig,
OMICRON electronics GmbH, Austria



**table 2 Functional components examples in UCA guideline (1997)**

| Device Model Definition: <Name> | | | | |
|---|---|---|---|---|
| FC | Object Name | Class (Data Type) | m/o | Description |
| MX | AV | AI (struct) | m | A Phase Voltage required |
| | AA | AI (struct) | o | A Phase Current optional |
| CF | AV | AOC (struct) | m | Config of A Phase Voltage |
| | - | | | |
| | - | | | |

Very soon the question for testing solutions for protection engineers occurred. OMICRON presented a test equipment to be used with UCA GOOSE in May 2001 at the IEEE PSRC Meeting with UCA Demo in Vancouver. Figure 4, Figure 5, Figure 6 and Figure 7 show the demonstration, Figure 8 shows the results.

So we learned that IEC 61850 is something like UCA 2.0 plus….

Why?

- There is a different terminology (Logical Nodes vs Bricks)
- The GOOSE is now expanded
- The control model differs
- There was something introduced called "Process bus" (to be covered later)
- Conformance testing is defined for the first time in a communication standard
- The general device requirements are defined as well as
- The communications requirements defined
- And finally there is a XML-based Engineering Language

Before we have to look into the details this paper shall summarize remarks form the early contributors. It is interesting to see that topics and terms still in use today occurred quite early.

Checking the early modeling guidelines there is a file issued in 1997:

Utility Communications Architecture (UCA), Version 2.0 Project. This document served as a complement to the RP 3599 Requirements Specification (Substation Integrated Protection, Control and Data Acquisition Requirements) and the UCA Application Services Model documents. The requirements for application modeling are defined clearly already:

- Easy of mapping to MMS and other protocols
- Self description
- Optional inclusion to allow vendors to select among choices for inclusion on certain components without affecting the interoperability
- Customization as the seed corn of future interoperable models
- Naming convention (CASM at this time vs. MMS Objects and Names)
- Device model documentation composition
- Common DataObjects
- Common representation of measured values
- Minimize complexity and depth of "nesting"
- Support grouping of information of common type and persistance

This document also introduces "functional components" which became later functional constraints (Table 2).

With GOMSFE (Generic Object Models for Substation & Feeder Equipment) the model could be described. A document released in 1997 shows Select Before Operate as control service as well as Report and Log Control Blocks as we know them in IEC 61850.

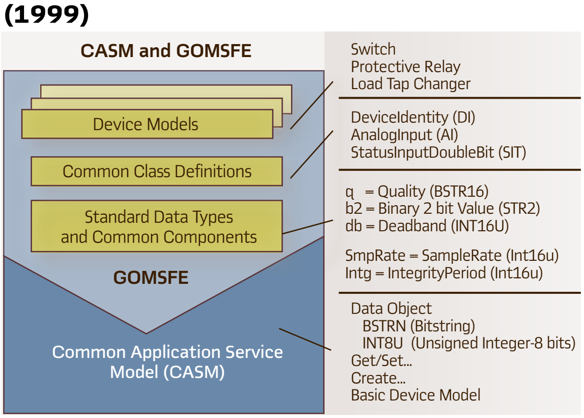
EPRI published the "Substation Protocol Reference Specification" to be used for "Substation Integrated Protection, Control and Data Acquisition" in the Utility Communication Architecture in 1997. Figure 9 shows the architecture.

Please note what "Real-Time Support" at this time meant.

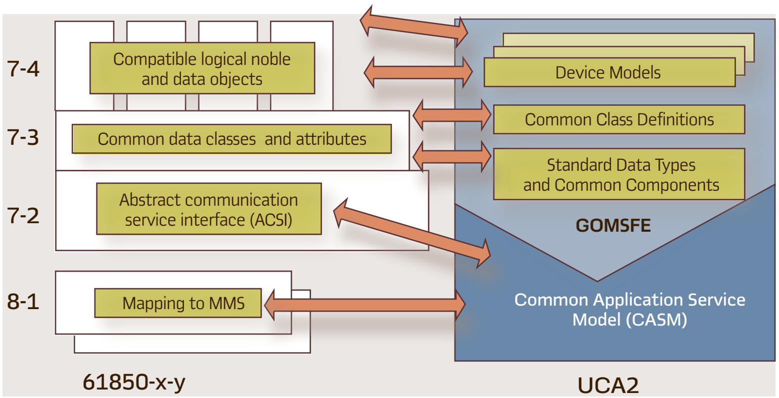
- It allows for automatic periodic or event-driven data (with a single "poll"), discard of old data (based on a time-allowed-to live).
- Data may be blocked to improve efficiency of messages.
- Applications handle the translation. (Presentation Layer is bypassed.)
- Priority data is defined.
- A performance fast stack (3 layers) is also defined.
- Time synchronization methods are provided.

The Substation Automation Communi-cations Demonstration Initiative presented the results in 1998.

**10 Archtitecture with GOMSFE and CASM (1999)**




**11 GOMSFE/ CASM and IEC 61850 (1999)**



■ Journaling provides for automatic logging of events or alarm conditions in real-time

This document from 1997 seems to be the first one describing "conformance blocks". Conformance blocks have been defined for simple, fast data exchange or more complex functions such as remote programs. This was necessary to distinguish between implementations for simple IEDs or more sophisticated processors at this time. There have been conformance blocks for real-time services, control, events and so on. Even time sync was mentioned already.

Time synchronization is an interesting chapter. It describes a method with time synchronization assuming a precision time source is available at the substation. The specification did not address the delivery of time to time masters at the substation location, e.g., by GPS or hardwired over dedicated lines or radio. The mechanism is recommended as an option on a subnetwork.

It provides a means to correct for queuing and other delays and imposes no requirement that time sync. messages be generated at precise times or intervals and was called "backwards time based correction" and worked as follows:

■ One device is designated as the "time master"

■ The time master is configured to send clock synchronization sequences at some interval called the "clock sync interval"

■ When the clock sync interval elapses, the time master broadcasts a message called a "timer event message"

So precision time protocol (PTP, today IEEE 1588) was defined. Until today technical issues are collected a tissues. The first time the word "tissue" was mentioned seems to be in a document from 1997.

The Substation Automation Communications Demonstration Initiative presented the results in 1998.

All the main parts of the UCA2-specification have been considered in IEC 61850 development. Of course some additions have been made by IEC. Also TASE.2 became a part of UCA2.

It was important for IEC to consider the 2 main parts of UCA2- the CASM and GOMSFE. Figure 10 shows the architecture. CASM described the functions for supervision, control and self description of IEDs. So the functions of CASM (Get, Set, Create, Reporting, Logging, Select-before-operate, ...) and the guidelines for modeling have been the base for GOMSFE. GOMSFE consisted of 3 layers. The first one defines the main data structures, necessary in any application. Additionally frequently used data structures and application specific properties will be defined. As mentioned we have structures for measurement or status information. The top level describes the information objects.

Figure 11 shows the terms in IEC 61850.

Guided by American Electric Power (AEP, one of the biggest utilities in the US) the "AEP Substation Initiative" was founded. Additionally by IEC OCIS (Open Communications in Substations) was initiated in 1998. Table 3 shows the utilities involved, Table 4 some active vendors.

Every vendor had to work with some utilities (Table 5) to achieve quick results together. During these projects a lot of new ideas came up and have been developed in products afterwards. We will cover this when writing about vendors and projects in the next issues.

The first IEC 61850 substation was put into operation in November 2004. But we are going to finalize this article with a rhyme created together at WG 10, 11, 12 meeting in Valencia (Spain) in 2000 (see page 70.)

walter.schossig@pacw.org www.walter-schossig.de

## table 3 AEP Initiative Utilities (1998)

| Company | State / Country |
|---|---|
| American Electric Power (AEP) | Columbus, Ohio (OH), USA |
| Maltimore Gas & Electric | Baltimore, Maryland (MD), USA |
| Boston Edison Company | Boston, Massachusetts (MA), USA |
| Cinergy | Plainfield, Indiana (IN), USA |
| Commonwealth Edison | Chicago, Illinois (IL), USA |
| Duke Power Company | Charlotte, North Carolina (NC), USA |
| Florida Power Corporation | Maitland, Florida (FL), USA |
| GPU Energy | Reading, Pennsylvania (PA), USA |
| Indianapolos Power & Light | Indianapolos, Indiana (IN), USA |
| Northern States Power | Minneapolis, Minnesota (MN), USA |
| NUON - TB | The Netherlands |
| Ontarion Hydro | Toronto, Ontario, Canada |
| Sourhern California Edison | Irwindale, California (CA), USA |
| Tampa Electric Company | Tampa, Florida, (FL), USA |
| Tennessee Valley Authority | Chattanooga, Tennessee (TN), USA |

## table 4 AEP Initiative Vendors (1998)

| Company | State / Country |
|---|---|
| Harris Corporation | Calgary, Alberta, Canada |
| General Electric Company | Malvern, Pennsylvania (PA), USA |
| Tasnet, Inc. | Pinellas Park, Florida (FL), USA |
| ABB Power Automation & Protection Division | Allentown, Pennsylvania (PA), USA |
| Alstom | Hawthorne, New York, (NY), USA |
| Basler Electric Company | Highland, Illinois (IL), USA |
| ARG Schneider Automation | North Andover, Massachusetts (MA), USA |
| Siemens Energy and Automation, Inc. | Brookland Park, Minnesota (MN), USA |
| Bitronics, Inc. | Lehigh Valley, Pennsylvania (PA), USA |
| Cooper Industries, Power Systems Division | Franksville, Wisconsin (WI), USA |
| Landis & Gyr | San Jose, California (CA), USA |
| Doble Engineering Co. | Watertown, Massachusetts (MA), USA |
| Beckwith Electric | Largo, Florida (FL), USA |
| REPS, Inc. | Clearwater, Florida (FL), USA |
| Dranetz/ BMI/ Electritech | Knoxville, Tennessee (TN), USA |

## table 5 Manufacturers and Utilities in AEP Initiative (1999)

| Vendor | Utilities |
|---|---|
| GE/Multilin (Relay) | IP&L, AEP, OH, ComEd, Cinergy, BG&E, TE, GPU |
| Basler | AEP, NSP, TE, BG&E |
| Cooper | AEP, NSP, IP&L, BG&E |
| GEC | ComEd, NUON, BE, BG&E |
| Beckwith | FPC, Cinergy, TE, NSP, GPU, IP&L, BG&E |
| Tasnet | BE, ComEd, BG&E |
| SEL | OH, NSP, TE, IP&L, BG&E, FPC, Duke, TVA, Entergy |
| GE/Harris | TE, OH, Cinergy, IP&L, FPC, Ameren, Entergy |
| Siemens (Relays) | OH, NUON |
| Siemens (HMI) | NSP, OH, NUON |
| Bitronics | NSP, BG&E, BE, Ameren, AEP |
| ABB | Cinergy, TU, SCE, BE, GPU, BG&E, NSP, Duke, TVA |
| L&G | TVA, BG&E |
| Doble | GPU, BG&E, ComEd, FPC |
| Dranetz/BMI | NSP, SCE, TVA, BG&E |
| Modicon/Square D | NSP, TVA, BG&E, OH |
| GE (HMI) | GPU, BG&E, NSP, ComEd, IP&L, BE |