Substation Control and Monitoring Interoperability Centered Multi-Modelling

Ursula Alexandrovna, Alice Premiana
Institut Trempet UQAM

CP. 8888 Succ. Centre ville
H3C 3P8 Montréal (QC) Canada
Tél. 1-514-987-3000 (6117)
Fax : 1-514-987 8477
e.mail Maffezzini.Ivan@uqam.ca

Abstract

This paper describes a multi-model approach to the description of substation control and monitoring, based on the IEC-61850 standard. We present the modeling as a translation from an "engineering" model (the standard) to a "user model" (our stakeholder oriented models). After a brief presentation of the Hydro Quebec (HQ) context, the public utility where a new IEC-61850 compatible system must be built, we present the principle of the Logical Node, the standard's main element. We have created three UML domain models to help stakeholders understand the benefits of the new system. These contain different numbers of classes. Common classes do not have the same attributes and operations are shown only in one model.

The first model is intended for HQ standardization agents and is a synthesis of the IEC 61850 standard with modifications to the abstract class Basic Logical Node. Standardization agents must have a good general knowledge of all aspects without this knowledge necessarily being in-depth

The second model has been elaborated for functional engineers and presents a functional viewpoint. In this model, non-functional requirements (interoperability and security) are absent. The first and second models are not consistent and this lack of consistency is an element for further analysis in the future. The model for a breaker is presented to illustrate the fact that the stakeholder sees only what concerns him..

The third model is intended for system and maintenance engineers, and is a complement to the second model. In this model the system classes and optional attributes linked to interoperability are represented. The abstract class basic Logical Node is presented in detail in this model only..

A table summarizes the importance of the IEC-61850 standard's main elements with respect to the stakeholders. A fourth stakeholder (the requirements engineer) is introduced in the table. We will also highlight the fact that all these models are independent of computer science knowledge and thus can be produced by domain experts or by any other person who has a good knowledge of classification methods (sociologists, philosophers, etc.). Some concerns about seamless transitions are presented.

Key-words

IEC-61850, Interoperability, Logical node, Modeling, Non-functional requirements, Seamless

Introduction

After a brief description of the context we will present our rationale for choosing a multi-model approach. A very short presentation of the concept of the Logical Node found in the IEC-61850 standard will introduce the models developed for the main stakeholders: standardization agents, system and maintenance engineers and functional engineers. Then we present the problem relied to the interweaving on function and non-functional requirements. The conclusion summarize the objectives reached and presents some concerns about seamless transitions.

The context

Hydro Quebec (HQ) is a Canadian public utility whose main mandate is the production and transportation of electrical energy. Twelve years ago, HQ was already at the cutting edge with its distributed monitoring and control system (SCC-HQ[1]) based on OSI standard protocols. Because such a system has already been in place for so long and stakeholders have become very familiar with it, at HQ, unlike many other public utilities, it is not necessary to persuade them of the advantages of a distributed system with a 802.3 LAN. What is difficult, at HQ, is to justify changing to a new system which has the same physical and functional characteristics as the existing system and which offers, as its only advantage, an improvement in non-functional requirements. In a company where the technical staff has a real influence on managerial choices, and no technical decision is made without their involvement and agreement, it is important that every stakeholder be able to participate in the decision-making process with a lucid comprehension of what is at stake.

There are four categories of stakeholders in the SCC-HQ plan (standardization agents, functional engineers, system engineers, and software engineers) and they work in four very isolated departments. In addition, we must add that the age-old quarrel of "old versus new", between employees who have been there many years and those newly arrived, which is present every time someone proposes a change to the system, is particularly strong at HQ, where the system is 12 years old. The "new" employees, in order to create a place for themselves (either psychologically or in order to further their careers) over-emphasize the defects of the old system and exalt all new technology. In contrast, the affection that the older employees have for the system they have used for many years and are familiar with, leads to a certain psychological inertia towards any suggested change, causing them to see any change as either risky or downright bad. We have not doubled the number of models because of this "quarrel". Instead, we have considered "old versus new" as something that colors and affects our reaction to the questions raised with respect to models..

Multi-modeling approach

There is no perfect system and every system results from a trade-off of some kind or another. In order to guarantee that the trade-off springs from a rational choice and is not merely a question of "the squeakiest wheel gets the oil", it is important that those who will be participating in the decision-making process be in a position to clearly understand all the issues. In our project, given the short time interval allowed for the writing of the System Requirements Specification (3 months), it was important that the new concepts stem from the background and knowledge that the stakeholders already have. The stakeholders have different visions because of their training, cultural background and organizational status. It would not be helpful to attempt to impose on them an alleged "objective vision". This is why we have introduced several different models. It is very important for us to talk about multi-models rather than simply multi-views of one model. These views imply that there is something to see which maintains a unity regardless of the one who is observing, and that this "something" is stable from these points of view (just as the famous elephant). In contrast, speaking of different models (descriptions) implies that there is no elephant at all, and that each model describes as objectively as possible an aspect of the reality that must be automated, and that all the aspects are integrated into a unity which is functional and not an "object". The expression "multi-model" emphasizes the fact that the model is just that: a model designed to aid in the stakeholders' understanding and not an objective description of a unlikely reality.

[1] *Système de Contrôle et de Commande (*Control and Monitoring System).

IEC-61850 and Interoperability

Interoperability is a non-functional requirement, which has an impact not only on performance and implementation costs but also on specification complexity. In an organization such as Hydro Quebec, it is particularly important to reduce the complexity induced on specifications by interoperability because the decision makers are the technical staff (our stakeholders) and not the financial staff. In our case, the first step is therefore to simplify the standard on which the system is based.

IEC is finalizing the publication of fourteen Parts that make up the IEC-61850 standard series. The IEC-61850's main objective is to "achieve interoperability between the IEDs supplied from different suppliers"[1], where IEDs (Intelligent Electronic Device) are the physical programmable devices exchanging data on the substation local area network.

The standard's main principle**,** the one allowing it to approach interoperability systematically, is the concept of the Logical Node (LN). The LN is the smallest component (brick) used to build a function, and that can be allocated to distinct IEDs. To do so, the data exchanged between LNs are defined formally. Interoperability is thus linked to the fact that various LNs carry out normalized sub-functions which exchange data, the name, structure and meaning of which are fixed by the standard. The LN is, in practical terms, the main concept that all the stakeholders must master quite thoroughly.

IEC-6850 standardizes 82 LNs. LNs are aggregates of:

- *Data Objects*: they represent specific information on the LNs. The number and type of data objects are specific for every LN but four data objects are common to all LNs: mode, behavior, health and nameplate. Data Objects can be of the standard data type (Integer, Boolean, etc.) or of the structured type (Quality, TimeStamp, etc), which are the Common Data Classes (CDC) defined in section 7-3;
- *Data Sets*: a group of commonly used data objects;
- and three other elements facilitating reporting and logging.

The framework is a client-server framework. The Part 7-2 defines the Abstract Communication Service Interfaces (ACSI) that makes the client's requests independent from programming languages and from protocols stack. Parts 8-1 maps ACSI onto Manufacturing Message Specification (MMS) application protocol. This approach makes the extensions and adaptations generated by technological change easier, because ACSI protect application processes from changes. IEC-61850 can be understood and his requirements can be implemented without knowing MMS so in our models there is nothing concerning communication protocols.

We do not describe the protection functions even if they are essential for the IEC approach because the protection department at HQ decided against building a new system around the IEC-61850 standard[2]. The choice not to provide modeling protection resulted in reducing the number of LNs from 82 to 46.

Since the IEC-61850 can be considered an engineering model and because we decided to build four stakeholder (user[3]) oriented models, the translation from the original model implies, for every destination model, choosing the elements which will be presented and going into greater detail into the elements chosen. The main "elements" among which to choose are: Functions (Part 5), ACSI and class models (Part 7-2), Common Data classes (Part 7-3) LNs (Part 7-4), Data Objects of chosen LN (part 7-4).

The models are built with UML and published in HTML on a WEB site [4].

Part 7-3 of the IEC-61850 standard defines *Common Data Classes* and part 7-4 defines LNs: they are the nucleus of the UML conceptual diagrams. In the standard all the "elements that constitute" the classes are presented as attributes without consideration of the attribute type (standard as *Integer* and *Boolean* or structured as *Quality* and *TimeStamp*). We decided to list only the standard ones as attributes. For example,

[2] We can see the SCC as the trojan horse for the introduction of IEC-61850 at Hydro-Quebec. Just as three thousand years ago, when discussion and strength are powerless, we must resort to tricks.

[3] We borrow the expressions "engineering model" and "user model" from the domain of human machine interfaces : "User" of the standard and not user of the system, of course !

the data object *Controllable Double Point,* instead of having fourteen attributes has seven attributes and is an aggregate of seven objects.

For all the models we decided to change the representation of the Basic LN (the abstract class from which the "functional" LN are derived). In the Basic LN, we have left the four mandatory attributes, the attributes related to interoperability only, and put the seven optional ones in the derived classes. This choice allows us to simplify function modeling by "forgetting" the Basic LN attributes in nearly all the diagrams.

Standardization agents model

Standardization agents, the instigators of the IEC approach, are above all concerned with establishing rules for restricting the technological and functional choices of system and functional engineers. They must be acquainted with all the elements of the IEC-61850, even if they do not need to know the details. But to be acquainted with IEC-61850, even if the standard is well structured and well written, is not so easy[4]. The standard is more than 500 pages long and contains dozen of tables.

As a result of the feedback received after a six hours course on IEC-61850 and a great deal of discussion, the following decisions were taken:

1. LNs: All LNs presented in Part 7-4 were included without the four mandatory attributes of Basic LNs.
2. ACSI: From the thirteen models presented in Part 7-2 (ACSI), only two (LN et Data) were not included because from the point of view of their attributes they are of interest to the standardization agents only. They are therefore described in 1).
3. Functions: All the functions (Part 5) are described but without emphasis on System support and System configuration.
4. CDCs: All the CDCs were presented. To simplify CDCs, two generalizations were introduced.

It is important for the standardization agents to be familiar with the rules used to create the new LNs and Data Objects presented in Part 7-4.

Functional engineers model

The functional engineers that we are concerned about here are the engineers responsible for the control function and not the engineers responsible for primary equipment. The main concern of the functional engineers is that all the functions carried out by the old system are executed by the new system without any changes. From a practical point of view, only the answers to two questions are important:

- Are the functions defined in Part 5 of the Standard that have an impact on the operation of the SCC compatible with the old functions?
- Does every primary equipment data of the old system have at least one corresponding LN attribute?

The answers to these two questions are related because in the Standard, functions and objects are intertwined. That is to say, the LN attributes are used as elements to describe the functions. To map the old system data to the IEC-61850 LNs, we extracted a UML model of present day switches from the listings of the old system. The results were "strange" in comparison to the standard. At least twenty single points statuses of the old system did not have a corresponding attribute in the switch-related LN. What is the best way to include these points in the new model? In a new LN that reflect the physical structure of the old system or in a standard LN? If the choice is a "standard LN", which one? XCBR (the breaker) or CSWI (the controller)?

Figure 1 presents a model for HQ functional engineers, using a breaker. The HQ breaker is represented as a logical device[5].

---

[4] A minimum of two months, full time.

[5] A Logical Device is defined by IEC-61850 as an agregate of LN.

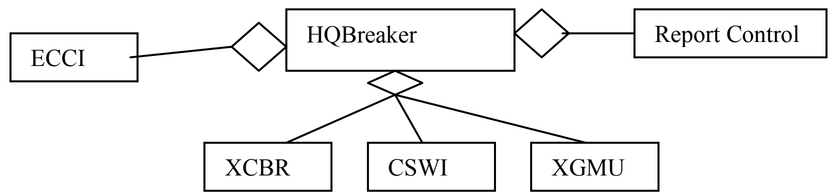

Fig. 1 : Breaker logical device

Four important points can be raised concerning this model:

1. Four LN switch related (XARC, XPDC, CPOW CILO) are absent. Why ? Because the functions associated are executed by specialized devices or because the function does not exist at HQ. This solution is not consistent with the standardization agents' model where all LN are present. Instead of trying of eliminate dissimilarities between the models by "forcing" the functional engineers to accept a more general solution, we preferred to keep the dissimilarities alive as an element for later study. At the beginning there was discussions about XGMU too. Why keep this NL if it has only one point for HQ ? Would it not be a better solution to put this "isolated" point with the others twenty ? In this case we decided that it was important to keep XGMU in the model because the respect of the standard imply that if an attribute exist you can't just introduce the same under another name just for… intellectual laziness.
2. The Report Control is directly attached to HQbreaker. This is not correct because in the standard the Report Control belongs to the LNs. Furthermore we do not differentiate buffered from unbuffered reports because at HQ all reports are buffered.
3. All logical devices must have a LLN0 (a logical node responsible xxxx). We don't represent LLN0 because it is of no functional interest. It is a solution to certain problems that arise as a result of interoperability.
4. ECCI is an extended LN introduced to contains the old system status points.

In summary: a) Because interoperability is not at all important to functional engineers (as far as they are concerned, it is simply noise), the system LN and all the features concerning interoperability were not presented in their model. b) Because they are interested in present day functions, we presented only the IEC-61850 elements concerning current HQ functions.

System and maintenance engineers model

The model for system and maintenance engineers is the complement of the model for functional engineers. The central elements are:

- Basic LNs with associated functions. These functions are not described in the functional model because at HQ operators are not allowed to control LNs;
- Security;
- System support functions (network management, time synchronization, etc.);

that is to say, all the ancillary elements that make exchanges between the logical nodes possible.

The reports, so important for functional engineers, were presented in this model only from the standpoint of flexibility and of their impact on perfective maintenance (the main reason for the introduction of the new system).

Table 1 summarizes the importance of the components described in the standard regarding the stakeholders. *Low*, *Ave*(rage), *High* qualify the importance of the component and *No* means not important at all. We have added the s/w engineer as fourth stakeholder, since it is he who will be the main person responsible for the activities that follow the modeling of the domain. Since our multi-modeled approach allows every stakeholder to concentrate on what he knows, the presence of a requirement engineer, whose main task is to describe and validate, is essential.,

	ACSI & classes	**ACSI Reports**	**CDC**	**Basic LN**	**Data Classes**	**Ancillary Functions**	**Primary Functions**
Stand. Agent	Low	Low	Low	Low	High	Ave	Ave
Funct. Eng.	Non	High	Low	Non	High	Low	High
S. & M. Eng.	Low	High	Ave	Ave	Low	High	Ave
Requir. Eng	High	High	High	High	Low	Low	Low

Table 1 : Stakeholders versus IEC-61850 components

The three models allow us to present the non functional requirements (often the requirement's Cinderella) without weighing down the functional part.

Non-functional requirements versus functional requirements

Even if the main concern of IEC-61850 is interoperability, two other non-functional requirements, security and performance, are thoroughly analyzed. If it is clear that performances are **non**-functional requirements (what is a performance if not a function attribute concerning time and space?), the separation between functional and non-functional requirements regarding interoperability and security is not so clear. IEC-61850 is a good example of the transformation of non-functional requirements into functional ones. In the case of interoperability, the functional and non-functional are so intertwined that the LN (the brick for interoperability) is, in a certain sense, a sub-function.. Security (a secondary non-functional requirement in IEC-61850) is described by an LN and two functions (System Security Management and Access Security Management). Availability is implicitly described by quality and the mandatory attributes of LN and explicitly described by the function *Physical device self-checking*.

IEC-61850 demonstrates well the fact that when a non-functional requirement is deeply analyzed it becomes a set of functions [3].

Loc as un example that mapping is not so easy

Loc is a data class that "indicates the switchover between local and remote operation" [1]. XCBR (the LN for the breakers) and CSWI (the LN for the switch controller) have a mandatory *Loc (Local Operation)* of type *SPS (Single Point Status)*. In the old HQ-SCC there was a point DIST indicating that there was a switchover. Are *Loc* and *Dist* the same thing ? They may or may not be. It depends on the point of view. First: *Loc* is an *SPS* and *SPS* is an aggregate with a mandatory attribute (Boolean) and two components: *Quality* and *TimeStamp* whereas *DIST* is a Boolean. From a functional standpoint, *DIST* is equivalent to the Boolean of SPS. But it is not equivalent from the standpoint of the SCC availability (*Quality*) and primary equipment maintainability (*TimeStamp*). Thus in the functional engineers' model we can "forget" *Quality* because it is a concern of system and maintenance engineers and *TimeStamp* because it is the main element of the *Events Recorder* Function (a function that was introduced and standardized for maintainability more than thirty years ago).

Supposing that *Quality* and *TimeStamp* can be "forgotten", can we map DIST to Loc ? But, which Loc ? The one of XCBR or the one of CSWI ?

A standardization agent put it into XCBR. His reasoning was "When DIST is true, the breaker can't be operated remotely, so…" But DIST indicates the physical position of a toggle switch of an electronic panel at the interface between the IED and the breaker and not the position of a toggle switch on the breaker. So we can consider DIST the equivalent of CSWI *Loc*. Yes, that's seems good… but, why not in a private LN as ECCI or… or…

From this simple example, beside the main lesson (mapping is difficult and there are multiple mappings), we have learned another lesson : the meaning of the data of an old system is often “written” into the code and the introduction of a new standard such as IEC-61850 can help to extract the meaning from the code and put it into the domain models.

Conclusions

From a non-functional requirement such as Interoperability follows a lot of functions and objects that make the building of a system more complex. This complexity is the price to pay for long-term economy. In this paper we have described an approach that transforms the “engineering model” of a very complicated standard (IEC-61850) into “user models”

The multi-model approach facilitates the comprehension of the new concepts by presenting a description similar to the “usual” model, the one everyone is already very familiar with. In addition, three secondary objectives can be reached:

1. To avoid pointless discussion about details the importance of which has been exaggerated with the sole aim of showing (not necessarily consciously) a thorough knowledge of the system.
2. To reveal, from the outset, the contradictions between the different models. These contradictions do not necessarily need to be eliminated when they are discovered as they allow a deeper, more thorough analysis of the domain.
3. To allow the software engineers to better grasp the different descriptions apart from the users' roles in use cases.

A multi-model approach before the System Requirement Specification allows the validation of all kinds of requirements, and not only functional ones. Such a validation comes before use cases, that is to say before the system delimitation. In a new complex system with strong dependability requirements, the improvement of validation compensates for the costs of producing various models. Such an approach also allows a consideration of functional and architectural characteristics which are deeply rooted in company policies, characteristics which are made obsolete by technology but which still seem necessary to the employees.

But, from the point of view of software engineering, the main indication of our experience concerns the analysis as a “special” domain. The building of “stakeholder-oriented” models allowed us to see in action the falseness of the “main truth” of object-oriented software: *In an object approach the transition from analysis to design is seamless*. We arrive at the same conclusion as Jackson in [1]: The seamless transition is only a consequence of a lack of analysis and of the confusion between system design and analysis. But, more radically than Jackson, we think that if the software engineers don’t leave the “domain” to the domain experts or to persons capable of conceptual organization (intelligent people with a first university degree, for example) the computerization will always be incorrect and *ad hoc*: That is to say with unforeseen maintenance costs. The presence of “software experts” during the analysis has also the perverted outcome of forcing the domain experts to act as software experts (or imitate them). In our case the simple fact of creating the models with a software engineering tool such as Rational Rose was a “disaster”. Instead of reasoning about “concepts”, the stakeholders began to speak about “classes[6]”, “methods”, “derivation”, “Data base”, all the software stuff that prevent the domain experts from describing the “true things” of the domain. Instead of describing what was important for the standardization of SCC functions, they behaved as software engineers (or programmers?) fighting with the domain with the "weapons" of the design.

References

[1] IEC, IEC-61850 Standard series : *Communication Networks and Systems in Substatio*ns

[2] http://trempet.uqam.ca/trempet/membres/Maffezzini/Télécommande/IEC61850/

[3] M. Jackson, *Problem Frames, Addison*-Wesley 2001.

[6]In the domain analysis “Concept” is less programming oriented than “Class” and therefore preferable.