E-Learning whilst eliciting and working IEC 61850 Substation, a case study

Ivan Maffezzini
Institut Trempet Université du Québe à Montréal
CP 8888 Succ. Centre ville
Montréal H3C 3P8
Canada
Maffezzini.ivan@uqam.ca

Alice Premiana
Institut Trempet Université du Québe à Montréal
CP 8888 Succ. Centre ville
Montréal H3C 3P8
Canada
premiana@labunix.uqam.ca

Abstract – **This paper describes how help and tutorial is integrated into a prototype of a configurator (SCALCID) for a substation control system (ALCID II) based on IEC 61850 standard. After exposing the context and requirements for an e-teaching system, two approaches are described. The first one is made up of a gradual integration of software requirements specification into the prototype via standardized templates. The second one is based on a four categories organization of help and tutorial documents: prototype, IEC 61850, substation control domain and modelling languages. The help and tutorial documents can be textual, audio and video. The conclusion presents some lessons that were learnt.**

I. INTRODUCTION

A. Context

Hydro-Quebec (HQ) is a Canadian public utility whose main mandate is the production and transmission of electrical power. At the end of 2000, TransÉnergie, the HQ division in charge of transmission, started a plan to standardize a new generation of protection and control system for substation and feeder equipments (ALCID II). The main difference — and virtually the only one — between the first generation of Hydro-Quebec protection and control system (ALCID I) and the second one is the introduction of an *interoperability requirement*. This interoperability implies that functionalities can be divided between several Intelligent Electronic Devices (IED) designed by different manufacturers. As a way to satisfy the interoperability requirements, the IEC 61850 standard was chosen [1]. IEC 61850 contains 10 parts, totaling up to 1,200 pages, and is quite complex because of the variety of languages used: Natural English, Structured English, XML schemas and UML.

ALCID II is a real time system that needs a very complex configuration to adapt the IEDs to the substation structure and dimensions. This paper intends to deal only with the configuration system, henceforth SCALCID.

B. Quality

From the requirements engineering point of view, the development of the real time system and of its configurator (SCALCID) must be approached in a completely different manner. The former is functionally stable and the only important new quality requirement is interoperability. The latter needs a totally new functional approach with stable quality requirements. But to retain at least the same level of quality of configurator as in the previous system (ALCID 1) it is of paramount importance to be careful about requirements engineering and in particular about verification and validation processes (V&V). Prototyping is the approach we chose to improve V&V. In particular we decided to focus on requirements elicitation and on proof of concept. One year was planned for the implementation of the prototype and we are now (November 2006) at the end of the first period dedicated to database validation.

B. Users

All SCALCID users are engineers and can be classified in two categories: standardization agents and functional engineers.

1. *Standardization agents,* Standardization agents, the instigators of the IEC approach, are mainly concerned with establishing rules for restricting the technological and functional choices. They must be acquainted with all the elements of IEC-61850, even though they do not need to know all the details.

2. *Functional engineers*. The main concern of functional engineers is that all the functions carried out by the old system are executed by the new system without any changes.

The standardization agents are IEC 61850 champions, but the ultimate decision to introduce or not ALCID II in a substation falls within the competence of functional engineers.

D. Transition

The transition towards ALCID II is possible if and only if the functional engineers have a good knowledge of the IEC 61850 standard and of the functions implemented into the configurators. Because of the complexity of IEC 61850 the “IEC 61850 champions”:

1. must introduce the standard “step by step” to steer clear of a mental block;

2. must hide all the peculiarities of IEC 61850 that are too far from the old system;

3. Must create a person-machine interface which is easy to learn.

II PROTOTYPE REQUIREMENTS

Since last May, *L'Institut Trempet* at the *Université du Québec à Montréal* is implementing a SCALCID prototype, based on an Oracle database. The original purposes of the prototype were:

1. Facilitating the requirement elicitation.
2. Validating the requirements;
3. Validating the database design.

When the person machine interface (PMI) is an important constituent of a system (as is the case in SCALCID), a horizontal prototype is generally supposed to validate the PMI. The task analysis mostly would allow the design of a PMI with optimized usability characteristics. As we think that a poor[1] PMI can help the learning we choose the opposite direction: the PMI was centered on the domain characteristics and not on the task analysis. In short: the validation of the PMI is far from being a requirement.

The main inputs to the prototype were:

1. A System Requirements Specification for the real time system;
2. A UML based specification of the domain;
3. The IEC 61850 part 6 standard.

The prototype is implemented in close collaboration with a standardization agent. The prototype implementation is based on the following hypothesis: *if the functional engineers do not master IEC 61850, they will never accept to introduce a new IEC 61850 based system.*

But how can the functional engineers have enough knowledge and enough know-how to master IEC 61850? The solution that would consist in "sending the functional engineers to a course" is not practical, not only because they cannot leave their jobs for a week or two (the minimal time required for learning such a complex standard) but above all because the mastery of the standard requires some practical work on the Hydro-Quebec environment.

For that reason we decided to add a new group of requirement to the prototype: *E-Learning for functional engineers.* Some key elements of the prototype that favoured an E-Learning approach are the following:

1. A centralized database is accessible via internet;
2. All the elements concerning the security and data property are integrated into the prototype;
3. The prototype is being developed in a teaching environment.

A. E-Learning requirements

In our opinion, the integration of working and learning is the most effective way to learn for the majority of functional engineers that are a decade or more moved away from the university benches.

To do so the following requirements were added to the prototype.

1. *E-01: Accessibility*. There must be 24 hours/day access to the prototype to allow learning-working even when the engineer is at home. The engineer must be able to work with the prototype from the start even if the functions are not yet fully implemented.

2. *E-02*: *"True substation"*. The substation must be a "true" substation chosen between the HQ substations. Every engineer must have his own database describing his substation.

3. *E-03: Software Requirements Specification (SRS) Integration*. To make things easier in the learning process while eliciting requirements, the functional part of SRS must be a dynamic constituent of the prototype.

4. *E-04*: *Help and tutorials*. Help files and tutorials must be integrated since the beginning. The structure, content and style of the Help and Tutorial must be like the final ones, that is to say: the learning part is not prototyped (even if it is not complete).

In this communication we will consider only the E-03 and E-04 requirements.

III SOFTWARE REQUIREMENTS INTEGRATION

The first version of the prototype was "inspired" by the configurator of ALCID 1 and by one of the few 61850 IED available on the market. The subsequent versions were based on a kind on "analysis stepwise refinement": when the requirements elicitation shows the utility of a function, the function is integrated as a menu with a simple "not implemented" window followed by a short description of the function. After this first description in a "free" natural language (a French description), there is a second one, template-based. The template is the standard requirements used in a SRS with some additional fields.

Here is a relatively detailed description of the evolution of a menu item whose content was run by the functional engineer. We have chosen this example because it is practically the only one where the functional engineer was the "teacher" and the standardization agent the "pupil". In the prototype development this contribution of the functional engineers was magnified a little to enforce that the system was also their system and thus to make system acceptance easier.

A. Five steps toward the solution

To explain the usage of the prototype as an SRS support, the main steps of the evolution of a set of parameters menu will be shown.

1. *First step*. When the user clicks on the menu bar an information window appears with the following message:

[1] It is obvious that this consideration must be taken *cum grano salis.*

“*Not implemented. This menu allows the user to prepare the database to receive a new set of parameters for a substation*”.

After a meeting with the functional engineers the function associated with the menu was organized into six menu items.

2. *Second Step*. Every one of the six menu items where associated with an alert box with a message of the same kind as the previous one. In particular, the message for the first menu item was: “*Not implemented. This menu item allows user to keep an identifier for a new copy of set of parameters for a substation*.” After this version the functional engineer gives the developers a PDF file with the form that must be filled to “keep an identifier”.

3. *Third Step*. Instead of implementing the form — because of other tasks at higher priority — the click on the menu item caused the display of the requirement. To make the interaction with users easier, and therefore support the V&V process, the requirements are documented on a standardized form as if it was part of the SRS. Here are the most significant fields:

- *Project management related information*: **Necessity, Priority** and **Stability.** These concern the final product, not the prototype. The possibility for the functional engineer to give his opinion on the factors that have the greatest influence on scheduling is a way to strengthen his support. The requirements with High Stability, for example, will be the first to be implemented in the final product. [2]
- *Requirement engineering related information*: **Input**, **Pre-conditions**, **Post-conditions**, **Description**. The content of these fields was prepared by a functional engineer. In the **Description** field a link was established with the Hydro-Québec document describing the reservation form. **Origin, Derivated**. These fields describe the “parent” of the requirements and his children as in SysML [3]
- *Software engineering related information*: **Rationale, Conflict**. These fields facilitate the design of the final product and the early detection of conflicts (which does not necessarily means “conflicts resolution”). The conflict field, in particular, is a means to go deeper into the analysis.
- *Explicit interaction*: **Open questions.** This field was introduced to favor the interaction between developers and functional engineers and between functional engineers and standardization agents. As for the **Conflict** field, the answer does not have to be given quickly: it is the life cycle of the question that is more important. The duration and, above all, the sub-questions generated, are a means to deepen the problem understanding (the *conditio sine qua non* for requirements V&V),

4. *Fourth Step*. This step changes the order of appearance of the requirement and the form that must be filled by the user as an input to the function. In step three, the form to be filled cannot be filled because it is a read-only scanned PDF file. In step four, the form is programmed and can be filled by the user and the requirement specification can be displayed as linked document (contextual SRS help). The Hydro-Québec form has been modified (addition of two e-mail fields). The addition, documented only in the prototype, was a result of a new need expressed by the functional engineer.

B. Rationale

We took this approach (to incorporate the SRS specification in the prototype) for many different reasons, the most important being:

- To avoid poor requirements validation because of the difficulties of understanding when reading an assembled SRS.
- To make easier the introduction of a methodology which tries to separate requirement engineering from software engineering.
- To prepare a bid whose main document is a merging of SRS forms and prototype

IV HELP AND TUTORIALS

A. Structure

Help and tutorial (H&T) documents were organized into four main categories: 1) prototype H&T 2) IEC 61850 standard H&T, 3) substation control domain H&T, 4) modeling languages H&T. For every category, there can be three kinds of documents:

1. *Textual document*. Useful for the learning at office or for a deeper learning after listening to the audio document.
2. *Audio document*. Useful for home learning and for the first contact with the system.
3. *Video document*. Useful for home learning or group learning in a conference room at the office.

Often the textual documents and the audio documents aim at the same knowledge but the audio one is never a simple reading of the text: it is less formal than the textual one and, at least in the PHT case, it is a support for some kind of animation. Figure 1 shows a contextual Help associated to a form for managing the IEC 61850 data. One tab for each category of H&T is present. In the figure, the SCALCID tab is active and the textual document about the use of the “IEC data management” is shown. Two controls are present: the first for a tutorial, the other for an audio help. All the contextual help windows have the same fields: Purpose, Rationale, Remark and Description.

B. Prototype HT

The textual help is a standard Windows help (similar to the Word 2003 Help). The Audio goes with an animation of the mouse pointer and there is no video. Because the prototype hides IEC 61850 from the functional point of view but it does not hide IEC 61850 for learning, in almost all PHT documents, there is at least one link toward IHT. The prototype H&T will be the main input to the final user manual.

C. IEC 61850 H&T

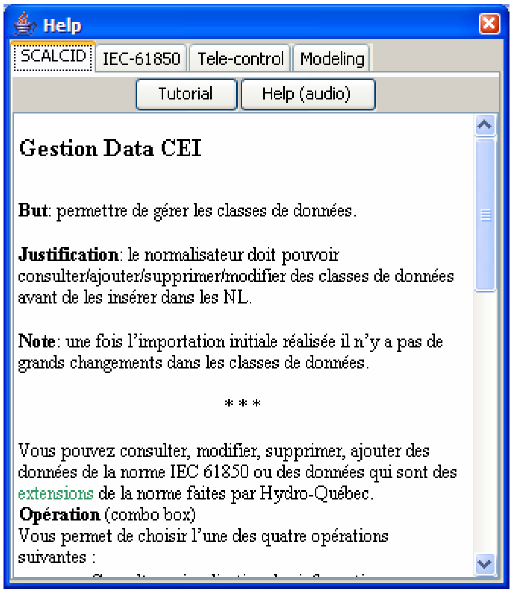
Because it was completely unrealistic to imagine a complete IEC 61850 H&T, our first choice was to introduce a few concepts that are at the foundation of IEC 61850 (as the IEC Data classes of Figure 1) and that were missing in the previous Hydro-Quebec system. The second choice was to explain the part of the standard which, in our opinion, is poorly integrated (see a detailed description in *F. Interaction between H&T categories*).

D. Domain H&T

Learning the domain is a low necessity and low priority task because the functional engineers, as functional engineers, must know very well the domain. So, why a domain H&T has been planned? Because some of the standardization agents are not so far away from retirement: the integration of their knowledge in the system is thus a way to make the integration of a new generation of engineers easier.

E. Modeling H&T

Knowing UML and XML schemas is a necessity to understand the IEC 61850-6, the part of the standard concerning configuration. Because of time and resources constraints, we decided to add to the prototype only a short description of UML classes and associations. For the final product a video tutorial will be realized.



Gestion Data CEI

But: permettre de gérer les classes de données.

Justification: le normalisateur doit pouvoir consulter/ajouter/supprimer/modifier des classes de données avant de les insérer dans les NL.

Note: une fois l'importation initiale réalisée il n'y a pas de grands changements dans les classes de données.

* * *

Vous pouvez consulter, modifier, supprimer, ajouter des données de la norme IEC 61850 ou des données qui sont des extensions de la norme faites par Hydro-Québec.
Opération (combo box)
Vous permet de choisir l'une des quatre opérations suivantes :

Figure 1. IEC Data classes management

F. Interaction between H&T categories

The power transformer parameterization is an instructive example of interaction between the four categories of learning material. Figure 2 shows a simplified UML description of a substation.

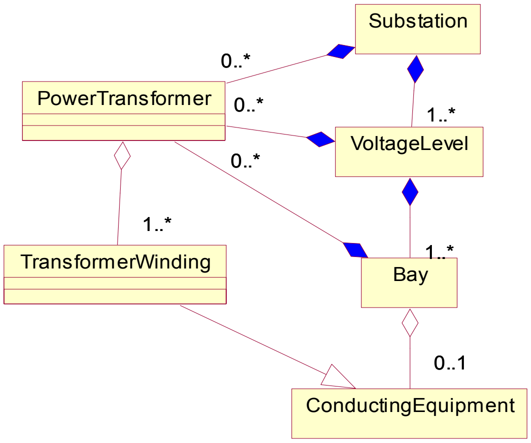


Figure 2. Power transformer and substation

A *Substation* is composed of one or more *Voltagelevel* and of 0 or more *Powertransformer*. A *VoltageLevel* is composed of one or more *Bay* and of 0 or more *PowerTransformer*. A *Bay* is an aggregate of 0 or more *ConductingEquipment* and of 0 or more *PowerTransformer*. A *PowerTransformer* is an aggregate of 0 or more *TransformerWinding* and a *TransformerWinding* is a *ConductingEquipment.* When we consider that the *TransformerWinding* must be a part of a Bay (as a special[2] *ConductingEquipment*) and, at the same time, a part of *PowerTransformer,* a difficulty of comprehension arises. To see the difficulty, we can consider the simplest significant case: a substation with two voltage levels, two bays (one for each voltage level) and one transformer with two transformer windings. One winding is in the first bay whereas the other is the second bay, at a different voltage level. But physically (and in the human representation) the windings are not detachable from a functional transformer (as our arms are not detachable from our body without transforming a trunk in a torso).

To clarify this oddness we choose to implement the power transformer parameterization in a doubtless awkward way. Contrary to what is the normal approach for a final product — to aim at a high usability — we did not hide any difficulties: the user must define the power transformer at the substation level, define the windings in the bays and then create a link as if the windings were not a constituent of the transformer. This lack of usability is a means to "force" the user to learn and, possibly at the same time, criticize the IEC 61850 modeling.

V CONCLUSION

A. Present situation

The introduction of IEC 61850 IEC in a substation brings about a lot of functional changes in the configuration

[2] *TransformerWinding* is special because it can have a *TapChanger*. We made some important changes to the IEC 61850 model to have a more easily interpretable model but the two models are semantically interchangeable.

consoles and a need for “cultural” change in the functional engineers approach – changes that can lead to rejection of the new system. A way to reduce the risk of rejection is to facilitate the learning of the standard while working at the specification of the new system. To do so, we developed a prototype with a rather sophisticated H&T system. A SRS is integrated by “stepwise refinements” into the H&T system and into the prototype.

Here is a synthesis of approach:

1. The scope of H&T goes from the contextual help for the prototype to tutorials for the modeling languages tutorials which, after some small modifications, can be reused in other domains.
2. Because of the easiness of programming, text, audio and video must be used as different media adapted to the users particular needs.
3. The H&T is driven by the questions of the developers and, above all, by the difficulties that the user finds in learning and working.
4. The requirements elicitation is done via the prototype and the requirements are integrated into the prototype as structured forms. No more SRS and prototype but SRS-prototype.
5. The requirement engineering as work in progress driven by the implementation of a prototype. NOTE. It is important to add that the prototype has already been used as a final product to standardize Bays and equipment. From this point of view the prototype is more a first version of the product than a prototype even if virtually all programs will not be reusable. END OF NOTE.

B. Next steps

Along with the stabilisation of the prototype the next main steps before the implementation of the product will be:

1. Introduction of controls into the SRS forms to allow the users to modify the content of the form and to validate it via a check list.
2. Introduction of a counter for the number of access to a specific SRS form as a way of improving quality assurance.[3]
3. Creation of the bulk of SRS from the SRS forms integrated into the prototype.

C. Lessons learned

Even if the more general lessons that we have learned form our prototype are far from being cut-and-dried, we dare to resume:

1. Learning while developing “together” a system is a facilitator of system acceptance and increases the system quality.
2. The prototype must integrate more flexibility than the final product to allow the engineers to prune the functions.
3. For an adult with a good knowledge of a domain, difficulties and oddness favour learning.
4. Trying to do several things at once in a homogeneous field of activity is a way to make a system more robust and easier to maintain.
5. Knowledge of the machine and knowledge of the domain are often so linked up that the efficiency of a clear cut separation — as so often trumpeted — is at least doubtful.
6. When a category of users is in a situation of “knowledge slavery” it is important to find at least a functionality where he is the “master” to give him some psychological self-confidence and, doing so, to diminish the risk of rejection.

There is certainly nothing new in theses lessons, above all for someone with some technical experience in developing complex systems and a bit of common sense. But they can have some interest in software engineering where a lot of people are still searching for the philosopher’s stone.

Our last remark is about the limits of our method (and of our lessons): a one year prototype for requirements elicitation and database design validation applies only to complex products with life cycle of several years. This remark about the limits of our method allows us to put an end to this paper giving utterance to a more general consideration of Michael Jackson about software engineering: “The idea of a panacea […] for the difficulties of software development is now rightly discredited”. [4]

VI ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of Pierre Martin, Myriam Say, Vincent Yu.

VII ACRONYMS

H&T:	Help and Tutorial.
IED:	Intelligent Electronic Device.
PMI:	Person Machine Interface
SRS:	Software Requirement Specification.
UML:	Unified Modelling Language.

VIII REFERENCES

[1] IEC 61850-6 *Communication networks and systems in substation – Part 6: Configuration description language for communication in electrical substations related to IEDs*,

[2] Ivan Maffezzini, “Most Stable Elements First Approach”, in *Proceeding of the 2006 3rd International Conference on Cybernetics and Information Technologies, Systems and Applications*.

[3] OMG, *SysML Specification - Final Adopted Specification*, May 2006.

[4] Michael Jackson, *Problem Frames*, Addison-Wesley, London: 2001, p. xvi.

[3] Suggestion made by Louis Martin software engineer and professor at UQAM.