

# Proposition pour gérer les défaillances conformément à la norme CEI 61850\*

Alice Premiana<sup>1</sup>, Pablo Martinez<sup>2</sup>

<sup>1</sup>Institut Trempet UQAM E-mail: [premiana@labunix.uqam.ca](mailto:premiana@labunix.uqam.ca)

<sup>2</sup>Institut Trempet UQAM -email: [martinez.pablo\\_daniel@courrier.uqam.ca](mailto:martinez.pablo_daniel@courrier.uqam.ca)

**Abstract.** This paper presents several approaches for managing the reporting of intelligent electronic devices (IED) failures in ways that are compliant with the IEC 61850 standard. After a short introduction to the standard and its functional division into logical nodes (LN), we shall analyse the IEDs and how we can deal with their failures in accordance with this standard. We shall finish this paper by recommending the introduction of a new logical node class and a group to contain it (as a short-term solution) and proposing the addition of a new part to the standard (as a long-term solution).

**Résumé.** Dans cette communication, nous présentons quelques approches à la gestion des défaillances pour des dispositifs électroniques intelligents (IED) conformes à la norme CEI 61850. Après une brève description de la norme et des fonctionnalités réparties parmi les nœuds logiques (LN), nous analysons les IED et comment les défaillances peuvent être traitées en conformité avec la norme. Nous terminons en proposant à court terme d'ajouter une nouvelle classe de nœud logique et un groupe pour la contenir et à long terme d'ajouter une nouvelle partie à la norme.

**Keywords:** Interoperability, IEC 61850, IED, failure.

**Mots clefs:** Interopérabilité, CEI 61850, DEI, défaillance.

## 1. CEI 61850 – MISE EN CONTEXTE

La norme CEI 61850, en définissant la structure et la sémantique des données qui transitent sur les réseaux locaux d'un poste, permet d'atteindre un haut degré d'interopérabilité entre les dispositifs électroniques intelligents (IED) qui se partagent les fonctions de contrôle/commande (CC). La norme définit l'interopérabilité comme « l'aptitude de deux ou plus IEDs d'un même ou de différents constructeurs à échanger de l'information et à l'utiliser pour exécuter correctement les fonctions spécifiées. » [1]

Les fonctions d'un poste sont réalisées par composition de sous-fonctions appelées nœuds logiques (LN). Les LN qui collaborent à l'exécution d'une fonction donnée peuvent appartenir à des IED différents. Les LN sont regroupés dans un contenant appelé dispositif logique (LD). La norme ne spécifie pas des LD normalisés, elle impose seulement qu'il y ait au moins un LD et qu'il n'existe pas de LN qui n'appartiennent pas à un LD.

À cause des idiosyncrasies de chaque système, il est pratiquement impossible d'obtenir une interopérabilité complète. Pour pouvoir tenir compte des particularités de chaque système, tout en gardant un bon niveau d'interopérabilité, la norme prévoit la possibilité d'ajouter, selon une structure fixée à l'avance, de nouveaux types de données dont la sémantique doit être définie dans un espace de nommage auquel les consoles qui configurent le système en temps réel ont accès. En exploitant cette ouverture, nous allons analyser quelques méthodes de gestions des défaillances des IED conformes à la norme pour le système ALCID II d'Hydro-Québec [2].

## 2. LES DÉFAILLANCES ET LE SYSTÈME

En nous inspirant de la définition de « *failure* » de la norme ANSI/IEEE 982-1 [3], nous définissons une défaillance comme « *La fin de l'aptitude d'un composant à exécuter une fonction dont il est totalement ou partiellement responsable.* » Mais comment respecter l'esprit de la norme (indépendance des constructeurs) quand on doit considérer la structure du matériel et du logiciel des IED, c'est-à-dire ce qui est propre à chaque constructeur ? Comment permettre aux responsables de l'entretien d'avoir une description assez précise du composant défectueux pour que le temps moyen de réparation (MTTR) soit réduit au minimum ?

Pour respecter minimalement l'esprit de la norme, il faut que la description permette de trouver le composant défectueux indépendamment du constructeur. Une façon de se libérer du « constructeur », c'est de proposer un modèle assez général pour intégrer tous les IED envisageables. La figure suivante présente le modèle à partir duquel nous analysons les défaillances.

Tout ce qui existe dans un poste et qui concerne les défaillances du contrôle/commande<sup>1</sup> est un module et chaque module peut avoir une ou plusieurs défaillances. Les modules sont dans une hiérarchie de type « tout-partie » (composition) avec le SCC au niveau 1 et les IED et les LAN au niveau 2. Les composants des IED (modules anonymes) constituent

---

<sup>1</sup> Même si nous nous limitons aux systèmes de contrôle/commande, nous pensons que nos considérations peuvent s'appliquer à tous les types de IED.

une hiérarchie qui commence au troisième niveau. Puisque tous les modules peuvent avoir plus d'un module de niveau inférieur, ce modèle permet de décrire les structures de n'importe quel IED : même la structure du logiciel où le terme « module » pourra indiquer une classe, un package, une fonction, un fichier, etc., selon les méthodes employées par le constructeur. À ce propos, il est important de souligner qu'un LN n'est pas un module mais une entité logique et que, donc, la défaillance d'un LN ne donne pas d'indication précise sur le module défaillant.

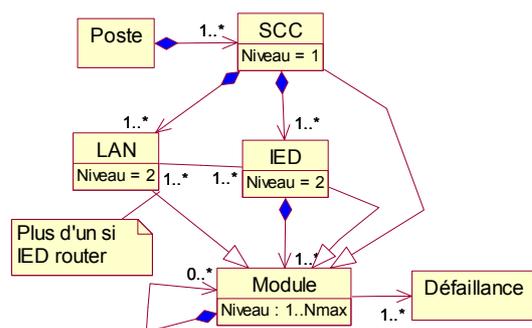


Figure 1 – Défaillances dans un poste

Dans cette communication, nous ne nous occupons pas de la manière de détecter les défaillances, mais seulement de la façon de les catégoriser, de les associer à un ou plusieurs modules et de les transmettre.

- **Catégorisation.** Nous nous inspirons de la catégorisation actuelle du système de contrôle/commande (SCC) d'Hydro-Québec (ALCID) [4] dont la deuxième version devra être conforme à la norme CEI 61850 [1].
- **Association.** Pour associer les défaillances aux composants (modules), nous nous appuyons sur le modèle de Figure 1 – Défaillances dans un poste.
- **Transmission.** Il s'agit d'intégrer les défaillances dans l'approche de la norme. C'est là l'objet de notre communication.

### 3. DÉFAILLANCES ET NORME CEI 61850

La norme ne traite pas des sources des défaillances mais se limite à :

1. associer à chaque donnée un attribut de qualité ;
2. déterminer un état de santé pour chaque LN ;
3. définir un nœud logique pour garder l'information de l'IED en tant que dispositif physique : *Physical device information* (LPHD) ;
4. commander l'exécution des diagnostics via un nœud logique spécial qui traite tout ce qui est commun à un LD (LLN0).

Il faut noter que la limitation dont nous parlons est une limitation du point de vue des techniciens d'entretien mais non du point de vue des opérateurs.

### 3.1. Qualité

À chaque donnée transmise par un IED est associée une information sur sa qualité. Deux attributs de la qualité nous intéressent particulièrement :

- *validity* qui peut être : *good, questionable, invalid, reserved* ;
- *detailQual* : qui détaille les causes de la condition *invalid* ou *questionable*. Parmi les causes, il y a *failure* qui indique « qu'une fonction de supervision a détecté une défaillance interne ou externe ». [1]

### 3.2. Tous les LN

Chaque LN contient une donnée (*Health*) qui donne son état de santé :

- *green* : aucun problème ;
- *yellow* : des problèmes mineurs mais les fonctions sont exécutées de façon sécuritaire ;
- *red* : des problèmes graves, impossibilité d'opérer.

### 3.3. LPHD

Il existe un nœud logique spécial appelé LPHD qui contient l'information du dispositif physique. Parmi ses attributs, il possède une donnée (*PhyHealth*) qui donne l'état de santé de l'IED. *PhyHealth* peut avoir les mêmes valeurs que la donnée *Health*, c'est-à-dire : *green, yellow, red*.

### 3.4. LLN0

Le nœud logique LLN0 centralise les informations sur les nœuds logiques qui ont été groupés dans ce que la norme appelle dispositif logique (LD). L'attribut *Diag* permet de lancer des diagnostics et peut donc être employé par un programme de supervision pour détecter des défaillances.

### 3.5. Conclusion

Tous les éléments que nous venons de présenter sont utiles et fondamentaux pour établir un niveau de confiance dans les données reçues et pour éventuellement déclencher une intervention d'entretien, mais ils ne sont pas suffisants pour satisfaire les besoins d'une console d'entretien à l'écoute du comportement des IED pour faciliter les interventions des techniciens. Nous allons proposer des approches qui devraient permettre de satisfaire les besoins d'entretien en temps réel pour améliorer la disponibilité du SCC.

## 4. BESOINS POUR L'ENTRETIEN

Pour améliorer la disponibilité d'un système, on peut agir sur le MTBF (Temps moyen entre pannes) et/ou sur le MTTR. Dans ce que nous proposons, il y a une intégration des défaillances à la partie temps réel des SCC pour améliorer le MTTR.

Les besoins pour l'entretien peuvent être groupés en deux catégories :

1. **État actuel.** Le SCC doit informer les clients de tous les changements d'état des défaillances dans un temps de l'ordre de quelques secondes. À l'état doivent être minimalement associés : sa gravité, le module directement concerné (source) et le/les modules des niveaux supérieurs de la hiérarchie qui sont affectés.

2. *Historique*. Le SCC doit conserver un historique des défaillances.

Dans cette communication, nous analysons seulement la partie « état actuel », car une fois les défaillances caractérisées et structurées pour l'état actuel, l'historique est facile à mettre en œuvre avec les reportages et les logs standard proposés par la norme.

## 5. SOLUTIONS CONSIDÉRÉES

Toutes les solutions que nous avons considérées sont fondées sur le fait qu'il est possible de définir (un)/des *dataset(s)* pour l'envoi des défaillances suite à un changement. Nous ne considérerons pas en détails les *datasets* mais seulement les éléments qui permettent de conserver une « photo » de l'état de santé des différents modules dans l'IED. Voici les six (6) solutions que nous avons envisagées :

1. **Modifier la définition du type qualité**. On pourrait ajouter des attributs ou modifier des attributs existants pour identifier le(s) module(s) défectueux. Le seul attribut qui peut être modifié est *source* qui, actuellement, peut valoir seulement *process* ou *substituted*. Il faudrait donc augmenter le nombre d'éléments de l'énumération en ajoutant toutes les sources envisageables. Puisque *source* est un attribut obligatoire de toutes les classes de données communes, sa modification rendrait très difficile la gestion de la compatibilité avec les versions précédentes. Ajouter de nouveaux attributs à la qualité entraînerait également des changements dans toutes les classes de données communes avec, en plus, l'ajout des frais généraux (*overhead*) pas forcément négligeables.
2. **Modifier le nœud LPHD**. Créer un nouveau LN qui spécialise LPHD en ajoutant des attributs pour représenter l'IED et inclure l'information des défaillances. Même si cette solution est conceptuellement simple et claire et s'inscrit très bien dans l'esprit de la norme elle pourrait impliquer un si grand nombre d'attributs que LPHD serait complètement dénaturé. Nous considérons une variante de cette solution dans la section 6, dans le cadre de solution à long terme.
3. **Modifier le nœud LLN0**. Il s'agit de modifier le nœud logique LLN0 en faisant une extension de cette classe pour lui ajouter la gestion des défaillances. Cette solution est une solution de compromis entre 1) et 2) qui n'a pas les qualités de 2) même si elle n'a pas tous les défauts.
4. **Utiliser GGIO**. GGIO est un LN générique (non lié à une sémantique du processus) qui permet de transmettre des informations analogiques et numériques et d'exécuter des commandes. La même norme recommande la réutilisation de ce nœud dans certains cas particuliers. Par exemple, dans la partie 5, on peut lire : « *Dès que les entités comme certains générateurs sont en dehors de la portée des sous-stations, mais qu'ils*

*possèdent quand même une interface de communication [...] ils doivent être décrits par un LN [...] comme le GGIO* ». Cette approche est pratiquement inacceptable car la fonction du GGIO est de régler des situations génériques et non pas des situations bien définies comme dans le cas des défaillances. Mais, même si on décidait d'employer les GGIO, il faudrait les spécialiser pour les adapter aux défaillances, ce qui les rendraient non génériques et donc... non GGIO !

5. **Modifier d'autres LN**. Les LN autres que LPHD, LLN0 et GGIO participent à l'exécution d'une fonction bien spécifique, il serait donc inacceptable de les transformer en agents pour les défaillances.
6. **Introduire un (des) LN dédié(s) aux défaillances**. Créer une (des) nouvelle(s) classe(s) LN dédiée(s) aux défaillances. Pour être cohérent avec la norme, cette approche doit introduire un nouveau groupe.

Dans le tableau suivant, nous présentons une synthèse des avantages des différentes approches. Pour chaque approche de la section précédente, nous considérons trois attributs : Facilité d'intégration dans la norme (FI), Facilité de gestion pour les IED (FG), Facilité de paramétrage (FP). Pour chaque facteur, on considère une échelle de 1 à 3 (3 étant la facilité maximale). La valeur a été fixée en discutant avec l'un des concepteurs du nouveau système d'Hydro-Québec. Même si les valeurs des attributs sont numériques, ils ne sont que des indicateurs qualitatifs.

**Table 1 Comparaison des approches**

|                                   | FI | FG | FP | Tot | Commentaire   |
|-----------------------------------|----|----|----|-----|---|
| Modifier l'attribut qualité       | 1  | 1  | 2  | 4   | On touche à la base même de la norme.                         |
| Modifier LPHD                     | 1  | 2  | 2  | 5   | Une variante de cette solution peut être viable à long terme. |
| Modifier LLN0                     | 1  | 1  | 2  | 4   | On change une fonction prédéfinie.                            |
| Employer GGIO                     | 1  | 2  | 2  | 5   | On rend les défaillances génériques.                          |
| Modifier d'autres LN              | 1  | 1  | 1  | 3   | Contre toute logique de la norme                              |
| Introduire un (des) nouveau(x) LN | 3  | 3  | 3  | 9   | Solution retenue  |

Dans la section suivante, nous allons détailler la solution retenue (à court terme) : introduction de nouvelles classes de LN.

## 5. SOLUTION RETENUE

### 5.1 Classes LN pour les défaillances

Nous proposons deux classes : l'une pour le matériel (FLHW, *Fault Hardware*) et l'autre pour le logiciel (FLSW, *Fault Software*). Le groupe F est actuellement disponible, donc, nous le réservons pour

les défaillances. La figure suivante montre l'intégration des nouvelles classes dans le modèle de la Figure 1 – Défaillances dans un poste.

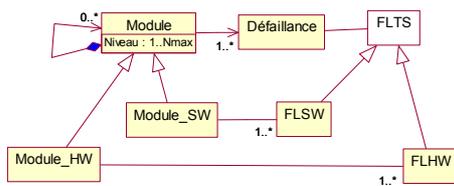


Figure 2. Modules et LN défaillances

Nous avons introduit la classe abstraite FLTS dans le seul but de rendre le schéma plus clair mais, dans la norme, il faudrait introduire seulement les classes concrètes FLSW et FLHW. Dans le tableau suivant, la classe FLHW est décrite selon le formalisme de la partie 7-4 de la norme, sans les quatre attributs obligatoires pour tous les LN.

Table 2 Classe FLHW

| Classe FLHW       |                |  |     |
|-------------------|----------------|--|-----|
| Attribut e Name   | Attr. type     | Description  | M/O |
| PhyHealth         | INS            | Doit être le même que <i>PhyHealth</i> de LPHD.  | M   |
| Filter            | INT16          | Durée minimale (en secondes) de l'anomalie pour qu'elle soit considérée comme disparue. Il s'agit de la valeur par défaut de toutes les défaillances.. | M   |
| ModHealth         | Health         | Green, Yellow, Red   | M   |
| ModDesc           | enum Module    | Module défaillant  | M   |
| ModPos            | PosCouple      | Position du module   | M   |
| Failure           | enum FailVal   | Défaillance trouvée  | M   |
| Date              | Timestamp      | Date d'apparition de la défaillance.   | M   |
| Cause             | enum CauseDef  | La cause de la défaillance (si elle est connue)  | O   |
| Possible Solution | enum Solutions | Possible solution pour la défaillance (changer la carte, etc.)   | O   |
| Descr             | VisString255   | Description détaillée pour faciliter le travail du technicien.   | O   |

FLSW et FLHW doivent être contenues dans un LD spécial et le nombre d'instances de ces NL sont prédéfinies dans l'IED (qui est censé connaître sa structure !) et sont décrites dans le fichier qui décrit les capacités de l'IED (fichier XML d'extension ICD).

NOTE : il serait sans doute préférable de créer dynamiquement les instances des NL des défaillances mais cette possibilité n'est pas prévue dans la version actuelle de la norme. Faudrait-il envisager un changement ? Sans doute, car s'il serait fonctionnellement très difficile (et sans doute inutile) de créer des instances pour les LN qui caractérisent l'installation, il n'y aurait aucun problème à créer des instances pour les LN des défaillances qui caractérisent seulement l'IED. En particulier, le

paramétrage des LN des défaillances n'a pas besoin des interventions des normalisateurs ou des projeteurs. FIN DE LA NOTE

Pour des « gros » IED cette solution peut impliquer un très grand nombre d'instances : dans les IED de contrôle/-commande actuels d'AICID, par exemple, elles pourraient être de l'ordre de quelques centaines. Pour pallier à cette « difficulté » nous avons considéré une solution où il existait une seule instance de la classe qui contenait un nombre de défaillances variables. Nous avons écarté cette solution car elle ne respecte pas l'approche par objets de la norme. Il faut dire aussi que la mise en œuvre d'IED conformes à la norme provoquera presque certainement la disparition des « gros » IED.

### 5.2 Type d'attribut

Pour faciliter la gestion des classes FLSW et FLHW nous définissons le type PosCouple comme suit :

Table 3 PosCouple

| Attribut type PosCouple |            |                                     |     |
|-------------------------|------------|-------------------------------------|-----|
| Attribute Name          | Attr. type | Description                         | M/O |
| Level                   | INT16      | Niveau dans la hiérarchie de l'IED. | M   |
| Position                | INT16      | Sa position dans le niveau occupé   | M   |

### 5.3 Énumérations

Voici un début d'une possible normalisation des énumérations :

- *FailVal* : NoFault, I/O, A/DConverter, NetworkCard, Memory, Chipset, NotNormalized, OptoBurntOut, Reserved...
- *CauseDef* : Unknown(0), HumanIntervention, HotTemperature...
- *Solutions* : Unknown(0), ReplaceCard, ReplaceMemoryStick, ReplaceChipset, RebootMachine...

La valeur *NotNormalized* dans *FaultVal* a été introduite pour ne pas rendre l'énumération trop longue. Dans le cas de défaillance *NotNormalized* l'attribut *Descr* est obligatoire.

### 5.4 Fonctionnement

Il est nécessaire de souligner que la qualité de l'information ne dépend pas de la transmission des défaillances telle que nous l'envisageons mais qu'elle dépend des mécanismes de détection et de mise à jour de la qualité des données des nœuds logiques déjà prévue dans la norme.

Dès qu'une défaillance est détectée (par exemple un opto-coupleur brûlé) l'attribut *FaultVal* de l'instance associée à l'opto-coupleur doit passer de *NoFault* à la valeur *OptoBurntOut*.

En fonction du type de la défaillance et de sa position dans la hiérarchie, l'IED pourrait aussi faire passer *FailVal* de *NoFault* à la bonne valeur dans les instances des classes FLHW des niveaux inférieurs de la hiérarchie. Par exemple, la défaillance d'une carte pourrait entraîner la défaillance de l'IED complet et, du point de vue de l'entretien, cela pourrait impliquer un changement de l'IED. L'état

actuel de l'IED serait donc caractérisé par l'existence de deux défaillances : carte (niveau 3 ou 4 en fonction de la structure du matériel) et IED (niveau (2). Lors de la détection d'une défaillance, si le technicien l'a prévu lors du paramétrage, l'information est stockée dans les fichiers de Log et/ou dans un *dataset* prévu à cet effet.

Le tableau suivant donne un exemple d'instance :

**Table 4 Instance de FLHW**

| Instance de la Classe FLHW |                |                          |     |
|----------------------------|----------------|--------------------------|-----|
| Attribute Name             | Attr. type     | Valeur                   | M/O |
| PhyHealth                  | INS            | Red                      | M   |
| Filter                     | INT16          | 30                       | M   |
| ModHealth                  | Health         | Red                      | M   |
| ModDesc                    | enum Module    | IOCard                   | M   |
| ModPos. level              | INT16          | 4                        | M   |
| ModPos. Position           | INT16          | 2                        | M   |
| Failure                    | enum FailVal   | OptoBurntOut             | M   |
| Date                       | Timestamp      | 2007/02/23 12 :34 :56.78 | M   |
| Cause                      | enum CauseDef  | Unknown                  | O   |
| PossibleSolution           | enum Solutions | ReplaceCard              | O   |
| Descr                      | VisString255   | opto-coupleur # 3 brûlé  | O   |

## 6. CONCLUSION ET TRAVAUX FUTURS

### 6.1 Conclusion

Dans cette communication, nous avons présenté quelques façons de gérer les défaillances des IED en conformité avec la norme CEI 61850. Après avoir analysé le pour et le contre des approches possibles, nous avons choisi d'ajouter une nouvelle classe de LN, avec 2 classes dérivées pour les défaillances matérielles (FLHW) et logicielles (FLSW) et un nouveau groupe, le F, pour les contenir. Nous avons donné des exemples de quelques énumérations qu'il sera nécessaire de compléter pour qu'une vraie interopérabilité puisse exister.

### 6.2 Travaux futurs

Le modèle de la Figure 2. Modules et LN défaillances, n'est pas sans analogies avec la structure du poste telle que définie dans la norme. Il serait donc logique d'ajouter à la norme une nouvelle partie (partie 11, par exemple) pour la gestion des défaillances. Par analogie avec les services *GetxxxDirectory* de la partie 7-2 de la norme (où xxx peut valoir : *LogicalDevice*, *LogicalNode*, *Data*, etc.), on pourrait avoir *GetxxxPhyDirectory* qui donne la structure physique de l'IED (où xxx pourrait valoir : *PhysicalDevice*, *PhysicalNode*, *Data* etc.). Comme la partie 7-4 présente les LN qui permettent de définir les fonctions pour le contrôle/commande et la protection du poste, la nouvelle partie 11 devrait définir les fonctions qui facilitent l'entretien en normalisant la description des IED. Dans cette approche il n'y aurait certainement pas deux seules classes de LN pour les défaillances mais autant de classe que de types de composants des IED.

## ACRONYMES

| Acronyme | Définition   |
|----------|--|
| FLHW     | Classe concrète pour les défaillances matérielles (elle étend FLTS).                                     |
| FLSW     | Classe concrète pour les défaillances logicielles (elle étend FLTS).                                     |
| FLTS     | Classe abstraite pour les défaillances.  |
| IED      | <i>Intelligent Electronic Device</i> .   |
| LD       | <i>Logical Device</i> .  |
| LN       | <i>Logical Node</i> .  |
| LLN0     | C'est un nœud logique spécial qui permet de connaître l'ensemble des nœuds logiques contenus dans le LD. |
| LPHD     | C'est un nœud logique qui permet de connaître l'état du dispositif physique.                             |
| SCC      | Système de contrôle/commande   |

## RÉFÉRENCES

- [1] IEC, IEC 61850 *Communication networks and systems in substations*, parties 1-10.
- [2] V. T. Nguyen, P. Martin, I. Maffezzini, *SES-02, Système de conduite d'installation, Spécification d'exigences de système*, Hydro-Québec, Février 2004.
- [3] IEEE, IEEE std. 982-1, *Standard Dictionary of Measures to Produce Reliable Software*, 1988
- [4] Guide des codes et libellés des défaillances du système ALCID/SICC (GT-XI-25B), Hydro-Québec, Décembre 1998

\* Nous remercions Jean-Louis Paquet (Cybetec-Cooper) et Pierre Paquin (Hydro-Québec) pour leurs critiques à la première version de cette communication.