

# Modèles Légués

Ivan P. Maffezzini et Brigitte Kerhervé

**Abstract**— Les systèmes légués ou systèmes patrimoniaux sont actuellement nombreux dans les entreprises. Ces systèmes, conçus et développés il y a de nombreuses années avec des technologies désuètes, nécessitent d'importants travaux de maintenance, que ce soit pour de la maintenance corrective ou encore perfective. Nous pensons que pour mener à bien un processus de maintenance d'un système légué, il est indispensable de se concentrer sur les modèles sous-jacents, modèles que nous définissons comme les *modèles légués*. Nous envisageons alors la maintenance de systèmes légués comme un problème d'ingénierie de modèles légués.

Dans cet article, nous revenons sur le terme *modèle* et sur les différentes utilisations et interprétations qui en sont faites. Nous définissons ensuite le concept de modèle légué et nous examinons les différentes formes que prennent ces modèles, à la fois dans la documentation mais aussi, et surtout, dans l'esprit des développeurs. Nous nous concentrons sur les changements menant à la maintenance des systèmes légués et nous illustrons nos propos à travers une étude de cas, à savoir la maintenance de systèmes légués chez Hydro-Québec, producteur, transformateur et distributeur d'électricité au Québec.

**Index Terms**— génie logiciel, modèles légués, maintenance.

## I. INTRODUCTION

Le génie logiciel (GL), tel qu'il est considéré par la majorité des chercheurs et des praticiens [1] est loin d'être une technique rigoureuse qui satisfait les exigences minimales de tout génie, c'est-à-dire « l'application d'une approche systématique, disciplinée et quantifiable » [2]. Pour s'en rendre compte il n'est pas nécessaire d'analyser la confusion qui existe dans la métrologie du logiciel [3] mais il suffit de considérer les changements de cap du GL en fonction de mots d'ordre plus ou moins à la mode. Mots souvent très ambigus et qui deviennent des espèces de talismans capables de régler les problèmes les plus dramatiques et coûteux de la discipline : de faire des vraies « révolutions ». Il suffit de penser à des termes comme *réutilisation*, *objet*, *re-ingénierie*, *exigences non fonctionnelles*, *processus*, *ingénierie des exigences*, *facilité d'utilisation*, *systèmes légués*, *développement piloté par l'architecture*, *cadre*, *interopérabilité*, *patrons* et, dernier de notre liste, mais non le moins important, *modèle*.

Université du Québec à Montréal (UQAM), Département d'Informatique  
 Courriel : [maffezzini.ivan@uqam.ca](mailto:maffezzini.ivan@uqam.ca), [kerherve.brigitte@uqam.ca](mailto:kerherve.brigitte@uqam.ca)

Le développement piloté par les modèles, par exemple, est présenté depuis quelques années comme la nouvelle révolution dans bien des articles. À ce propos, le numéro de septembre 2003 de *IEEE Software* est une bonne synthèse des recherches et des produits ayant les modèles comme noyau de base. Même si en [4] l'auteur se demande, justement, si le développement piloté par les modèles ne risque pas, comme les autres révolutions, d'avoir « des impacts minimes ou non fondamentaux », les autres articles ne sont pas si prudents.

La réponse à la question « qu'est-ce qu'un modèle ? » donnée en [5] ne nous satisfait pas complètement, c'est pour cela que nous nous sentons en devoir d'éclaircir ce que nous entendons par modèle avant d'aborder le centre de notre communication concernant les impacts des modèles légués sur la maintenance perfective.

## II. MODÈLE ET MODÈLE LÉGUÉ

### A. Qu'est ce qu'un modèle?

La liste des syntagmes où apparaît le terme *modèle* en GL est passablement longue. En voici une qui est loin d'être exhaustive : modèle objet, modèle conceptuel, modèle de données, modèle physique, modèle mathématique, modèle du système, modèle d'ingénierie, modèle environnemental, modèle abstrait, métamodèle ontologique, métamodèle linguistique, modèle sémantique, modèle utilisateur, modèle formel, modèle dynamique, modèle statique.

En lisant cette liste il est naturel de se poser une question du genre : « Est-ce que le terme *modèle* garde une signification stable en changeant de syntagme ou est-ce que le terme qui l'accompagne en bouleverse la signification ? » Question qui est préalable à toute tentative de classification des modèles.

Notre impression est que le terme modèle change tellement de signification qu'il vaut la peine de faire un retour aux sources, c'est-à-dire au langage ordinaire tel que présenté dans un dictionnaire avant d'essayer de dire ce qu'est un modèle pour nous.

Dans la langue française (comme dans la langue anglaise) le mot « modèle » est un mot avec une très grande polysémie. Le dictionnaire *Le Robert*, par exemple, propose neuf (9) acceptions différentes de *modèle*. En GL, heureusement, le terme modèle n'est pratiquement employé que selon trois acceptions :

1. *Ce qui sert ou doit servir d'objet d'imitation pour faire ou reproduire quelque chose.* Les patrons d'analyse et les patrons de conception en sont de bons exemples.
2. *Représentation simplifiée et plus ou moins formalisée d'un processus, d'un système.* C'est certainement

l'acception la plus employée et qui couvre aussi une partie du champ sémantique de 1), car la simplification facilite la reproduction.

3. *Modèle d'une théorie : structure dans laquelle tout énoncé de cette théorie est vrai.* Il s'agit de l'acception de la logique mathématique.

Que l'on n'emploie que ces trois acceptions n'implique pas que les autres ne puissent pas compliquer un peu plus les choses en ajoutant du bruit sémantique<sup>1</sup>.

Dans cet article nous ne considérerons que l'acception 2). Nous ne traiterons pas de l'acception 1) parce qu'elle sera considérée comme étant couverte par la 2).

La deuxième acception est une définition satisfaisante pour un dictionnaire de la langue française mais elle a besoin d'être détaillée et précisée dès que l'on doit l'employer dans un travail technique ou scientifique. Comme il est « naturel » pour une langue naturelle (!), cette définition est à son tour polysémique avec des significations qui s'opposent sinon du point de vue sémantique, certainement du point de vue pragmatique.

Si on prend la physique comme modèle d'une science exacte (ici *modèle* est employé dans l'acception 1) précédente) on s'aperçoit qu'en physique aussi il n'y a pas qu'une signification de « modèle ». Dès le lycée, par exemple, on présente le modèle atomique de Bohr<sup>2</sup> comme un modèle (dans l'acception 2) pour comprendre la structure des atomes. Le modèle de Bohr est une amélioration de celui de Rutherford qui à son tour était une amélioration de celui de Thompson qui, si l'on veut, était une amélioration de celui de Lucrèce.

Indépendamment de ce que nous avons appelé amélioration, qu'est-ce qu'ont en commun ces modèles ? Ils ont en commun le fait que la « *Représentation simplifiée* » est *inspirée par analogie de la réalité physique*. L'atome selon le modèle de Rutherford, par exemple, n'est qu'un système solaire en miniature. Le modèle de Bohr, même s'il introduit des changements théoriques radicaux, reste fortement lié au modèle planétaire. Ce que l'on a appelé « amélioration » n'est qu'une adéquation toujours plus grande du modèle aux objets externes, c'est-à-dire, dans ce cas-ci, aux constituants élémentaires de la matière. On peut parler d'une plus grande adéquation parce que, via la technique, on est à même d'interagir avec la matière de manière plus efficace.

Ce genre de modèle facilite la compréhension des humains des phénomènes sous étude parce qu'il exploite celle qui est sans doute l'une des caractéristiques les plus puissantes du cerveau humain : le raisonnement analogique. On peut donc appeler ces modèles des modèles analogiques.

Mais le raisonnement analogique, si puissant pour la compréhension, est déficient en ce qui concerne les traitements quantitatifs. C'est pour cela qu'en physique le

modèle de Bohr est « accompagné » par des équations qui décrivent quantitativement les phénomènes et qui permettent de faire des mesures (souvent statistiques) pour faire des prévisions. Ces équations sont, elles aussi, un modèle : c'est-à-dire une *Représentation simplifiée* de la réalité. Nous appellerons ce modèle un modèle analytique. C'est la confusion entre ces deux types de modèles qui crée parfois une grande confusion en GL [6].

En GL les deux types de modèles, analogique et analytique, sont importants et complémentaires : sans le premier il est difficile, sinon impossible, de synthétiser les connaissances d'un domaine<sup>3</sup>, sans l'autre il est pratiquement impossible de mesurer et de faire des prédictions. Le premier type de modèle est surtout utile lorsque l'on veut comprendre le problème et les besoins que le problème induit, le deuxième est utile pendant toutes les phases et son poids plus ou moins grand dépend du domaine que l'on veut informatiser.

### B. *Qu'est ce qu'un modèle légué ?*

Dans la littérature concernant la maintenance on traite souvent des systèmes légués (*legacy systems*), c'est-à-dire des « vieux » systèmes que des ingénieurs du logiciel laissent en legs et qu'il faut entretenir d'une part pour corriger les défauts (maintenance corrective) et d'autre part pour les adapter à de nouvelles exigences (maintenance perfective).

L'entretien de ces systèmes est souvent très coûteux sinon impossible. Ceci est moins vrai pour des systèmes qui commencent à être « vieux » mais qui ont été développés dans les années 80 selon des pratiques de GL « correctes ». Ces systèmes<sup>4</sup> sont documentés, parfois « bien » documentés et peuvent être caractérisés par trois types de modèles qui préexistent et qui sont légués avec la machine :

1. Les modèles explicitement décrits dans la documentation. Il s'agit souvent de diagrammes de classes ou de diagrammes hiérarchiques accompagnés de descriptions plus ou moins formelles. Ces modèles peuvent être analogiques ou analytiques.
2. Les modèles que l'on peut induire de la documentation<sup>5</sup>. Il s'agit de structures que l'on peut assez facilement extraire de la documentation en langue naturelle. Le modèle est pratiquement toujours analogique. On peut imaginer extraire un modèle analytique mais avant d'arriver au modèle

<sup>3</sup> Même si les modèles analogiques sont moins précis, les physiciens en ont besoin car la compréhension via les équations mathématiques est une « compréhension » désincarnée qui ne permet pas de comprendre comme on comprend dans la vie quotidienne. Il serait sans doute préférable d'utiliser un autre nom pour nommer la compréhension à l'aide des équations afin de mieux la différencier de la première.

<sup>4</sup> À ce propos il faut ajouter qu'un système informatique qui fonctionne correctement est lui-même un modèle de ce qu'il informatise. Nous n'aborderons pas cet aspect de la modélisation.

<sup>5</sup> Il y a aussi les modèles que l'on peut construire en partant des listages. Mais ce cas-ci est le cas « classique » de rétroingénierie des systèmes légués que non seulement nous n'aborderons pas mais par rapport aux fondements duquel cette communication s'inscrit, au moins partiellement, en faux.

<sup>1</sup> À titre d'exemple, voilà deux autres acceptions : 1) « *Objet, type déterminé selon lequel des objets semblables peuvent être reproduits à de multiples exemplaires par l'industrie* ». 2) « *objet possédant au plus haut point certaines qualités ou caractéristiques qui en font le représentant d'une catégorie* ».

<sup>2</sup> Appelé aussi modèle de Bohr/Schrödinger.

analytique il faut, au moins dans sa tête, faire un détour par le modèle analogique.

3. Les modèles « conceptuels »<sup>6</sup> dans la tête des individus. Il s'agit des « représentations » abstraites du problème, du domaine, du système, etc. que les individus se sont construits tout au long de la vie du produit. Ces systèmes ne sont pas assez « vieux » pour que certaines personnes qui étaient présentes au début n'interviennent pas à nouveau dans le projet. Il s'agit là de modèles analogiques.

Ces modèles, chacun à sa façon, influencent la progression des travaux lorsque l'on doit opérer des changements importants dans la conception du système et, surtout, quand on doit le refaire.

Dans la prochaine section nous considérons un sous-ensemble des modèles légués du système ALCID, un système de contrôle et de commande des installations électriques d'Hydro-Québec.

### III ETUDE DE CAS

#### A. Le système

En 1985 Hydro-Québec, la société responsable de la production, du transport et de la distribution de l'énergie électrique au Québec, lança un projet de modernisation de la conduite de ses installations. Il s'agissait de substituer les stations terminales (RTU *Remote terminal Unit*) responsables de la commande des équipements et de l'acquisition des données d'une installation par un système distribué. Le système proposé (ALCID) est structuré autour d'un réseau local CSMA/CD doté d'une couche transport de type ISO-8073. Les fonctions d'un système de conduite peuvent être classées en deux catégories [7] :

1) *Fonctions temps réel* : il s'agit des fonctions qui permettent de réagir aux événements dans l'installation et aux requêtes des opérateurs en des temps assez courts pour ne pas perturber le fonctionnement. Elles sont regroupées en :

- *Conduite* : fonctions responsables de la commande et de la supervision des appareils incluant la gestion des alarmes et des mesures.
- *Automatismes locaux* : fonctions qui automatisent les procédures de manœuvre des appareils, déclenchées par un ou des événement(s) survenu(s) dans l'installation.
- *Comportement* : fonctions qui enregistrent des événements horodatés (avec une précision d'une msec) et les fournissent à un centre de contrôle pour l'analyse post-mortem des incidents.
- *Support* : ces fonctions assurent une qualité de service acceptable surtout en termes de disponibilité

<sup>6</sup> Il s'agit du modèle « dans la tête » et non du modèle du domaine éventuellement décrit dans la documentation.

et de sûreté.

2) *Fonctions d'ingénierie* : il s'agit des fonctions qui permettent aux ingénieurs de configurer le système pour l'adapter à une installation particulière. Ces fonctions sont groupées en :

- *Gestion de la base de données* : fonctions responsables de la mise à jour de la base de données de configuration, de sa vérification et de son chargement dans les IED (*Intelligent Electronic Devices*) responsables de fonctions temps réel.
- *Gestion des automatismes* : fonctions responsables de la compilation, de la mise au point, du chargement et du déverminage des automatismes.

La figure suivante présente les acteurs qui interagissent avec le système ALCID.

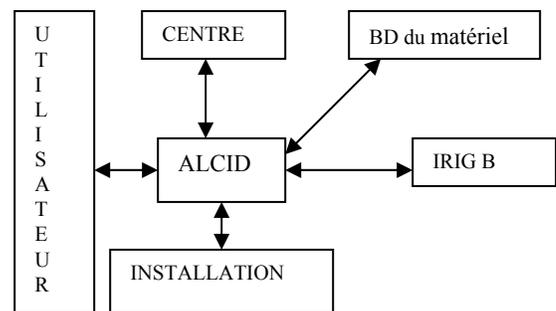


Figure 1. Acteurs du système ALCID

Où :

*Utilisateur* : 1) utilisateurs du temps réel : opérateurs dans l'installation et les techniciens d'entretien. 2) utilisateurs pour le paramétrage : ingénieurs.

*Centre* : il existe trois types de centres qui peuvent commander ou surveiller l'état des installations.

*Installation* : contient les appareils qu'il faut contrôler et surveiller.

*BD du matériel* : base de données qui contient la description des caractéristiques des équipements.

*IRIG - B* : date, heure de synchronisation.

La première version du système ALCID fut installée en 1989. Le développement a été réalisé en s'appuyant sur un sous-ensemble des normes du génie logiciel d'IEEE. En 1999, à cause de la difficulté d'approvisionnement du matériel et des coûts de maintenance perfective, un projet de modernisation d'ALCID a été lancé. La facilité de maintenance et l'interopérabilité devinrent les exigences principales à la base du nouveau projet. La modélisation du domaine et la spécification des exigences système ont été livrées en 2003 et une première version de la conception de la base de données en 2004.

Il est important de souligner que quatre personnes impliquées dans le projet en 2004, l'étaient aussi en 1986.

Dans cette étude, à titre d'exemple des impacts des modèles légués, nous considérons la modélisation des disjoncteurs. L'historique des modèles sera suivi par des considérations sur

les difficultés d'évolution.

### B. Modélisation des disjoncteurs

Les disjoncteurs sont des équipements primaires (voir la définition en *Annexe 1*) essentiels pour le bon fonctionnement d'une installation même s'ils ne produisent pas d'énergie (comme les alternateurs), ne changent pas les niveaux de tension (comme les transformateurs), ni la transportent (comme les lignes). Leur fonction est d'interrompre le courant (à cause de défauts ou par des choix d'exploitation) en s'ouvrant, ou d'en permettre le passage en se fermant. Ces opérations logiquement si simples sont physiquement très compliquées et nécessitent beaucoup de surveillance. C'est pour cela que l'état du disjoncteur n'est pas un état avec deux seules valeurs possibles (*On*, *Off*), mais un état à quatre positions : *En mouvement* et *Défaut* s'ajoutent aux deux valeurs de « base ». Dans pratiquement tous les systèmes de contrôle/commande, indépendamment de la technologie utilisée, l'état du disjoncteur est représenté par deux bits.

TABLEAU I  
REPRÉSENTATION DE L'ÉTAT D'UN DISJONCTEUR

Bit 0	Bit 1	Signification
0	0	Ouvert
0	1	En mouvement
1	0	Défaut
1	1	Fermé

A cause de la sûreté de fonctionnement exigée, l'opération de fermeture ou d'ouverture est bien plus compliquée qu'un simple changement d'état d'un bit d'une carte de sortie. Souvent, il faut en premier sélectionner un relais (sélection) et ensuite commander le changement d'état (exécution). Entre la sélection et l'exécution une signalisation est envoyée vers l'IED responsable de la commande pour vérifier le bon fonctionnement des cartes de sortie.

De 1985 à aujourd'hui, la modélisation des disjoncteurs a grandement évolué. Nous présentons ici les six grandes étapes de cette transformation en caractérisant chacune d'entre elles.

1. Modèle « *Points d'Entrée/sortie* ». Le point de départ du projet ALCID consistait en la documentation existante décrivant les RTU. Or, dans cette documentation, le concept de disjoncteur n'existe pas. Tout est centré autour des points d'entrée et des points de sortie. Un disjoncteur est réduit à deux points d'entrées (les deux bits du tableau précédant) et deux points de sortie (sélection et exécution). Mais le fait que ces 4 bits identifient un disjoncteur n'existe que dans la tête des ingénieurs (modèle implicite). Dire qu'un « disjoncteur est réduit » est un abus de langage, au moins du point de vue de la documentation : comme nous venons de le dire il n'existe pas de disjoncteur explicitement décrit. Ceci est clairement visible dans les consoles de paramétrage où il n'existe même pas de modules (du code, donc) concernant les disjoncteurs, mais seulement des modules pour caractériser « ses » point. Dans la partie temps réel on trouve des algorithmes qui « lient les points entre eux » et qui

constituent un modèle exécutable du comportement des disjoncteurs (il s'agit d'un modèle analytique, pas très utile à la compréhension car il n'y a pas de modèle analogique).

2. Modèle « *universel* ». Le premier modèle d'ALCID est le même que celui des RTU. Suite aux difficultés de paramétrage des installations dotées d'ALCID, en 1989 il est proposé un modèle très général inspiré des spécifications électriques du disjoncteur. Pour la première fois un modèle explicite de disjoncteur est donc introduit. Le disjoncteur est considéré comme une spécialisation d'équipement qui a les attributs suivants : *nom du constructeur*, *marque*, *No de série*, *symbole pour l'affichage*. Le disjoncteur devait exécuter des messages du genre: *Affiche-toi*, *Ferme* et *Ouvre*. Il n'y aura qu'un prototype vite abandonné qui met en œuvre ce modèle jugé trop complexe par rapport aux besoins du contrôle/commande.
3. Modèle « *container* ». L'année suivante le modèle précédant est simplifié en laissant tomber un ensemble d'attributs et d'opérations jugés non nécessaires pour le contrôle commande. Huit types de disjoncteurs sont introduits comme des spécialisations d'équipement et ils se distinguent selon s'ils peuvent être commandés à distance et s'ils peuvent envoyer des signalisations. Les types de disjoncteurs sont : *Manuel*; *Signalé*; *Monopolaire télécommandé*; *Monopolaire télésignalé*; *Tripolaire télécommandé haute tension*; *Tripolaire télésignalé haute tension*; *Tripolaire télécommandé haute tension*; *Tripolaire télécommandé moyenne tension débrouachable*.  
Ainsi, un disjoncteur *Manuel* est un disjoncteur qui ne peut pas être commandé via ALCID et qui n'envoie pas de signalisations; alors qu'un disjoncteur *Signalé* est un disjoncteur qui ne peut pas être commandé via ALCID mais dont les signalisation sont rapatriées. Chaque type de disjoncteurs est caractérisé par des points d'entrée et de sortie pouvant aller d'un minimum de 2 points pour le disjoncteur manuel jusqu'à une trentaine pour le disjoncteur monopolaire télécommandé. Chaque point des disjoncteurs est caractérisé par :
  - a. *Nom* : nom en clair du point.
  - b. *Type* : entrée numérique, sortie numérique, entrée analogiques, sortie numérique.
  - c. *Envoyé à*
    - a. *Centre 1* : booléen
    - b. *Centre 2* : booléen
    - c. *Etc.*
  - d. *Fonction* : énumération des fonctions associées au point.

La Figure 2 synthétise ce modèle :

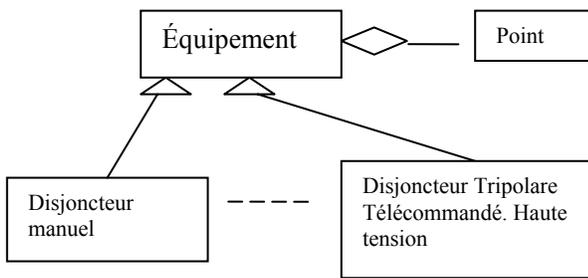


Figure 2. Modèle Universel

4. Modèle « hiérarchisé ». Dans le cadre du nouveau projet de 1999 on introduit une nouvelle modélisation pour répondre aux préoccupations principales qui sont l'interopérabilité et la facilité de maintenance parfaite. Dans la modélisation précédente les disjoncteurs n'étaient, au fond, que des containers de points sans hiérarchie, ce qui rendait difficile la compréhension des différences entre les disjoncteurs. Dans le but de mieux comprendre le système existant, il fut donc décidé de transformer la hiérarchie précédente à deux niveaux en une hiérarchie à 7 niveaux avec l'introduction de quelques classes abstraites. On décida aussi de ne pas séparer les autres appareils de coupure (sectionneurs et interrupteurs) des disjoncteurs. La Figure 3 présente le premier d'une série de trois diagrammes UML représentant les appareils de coupure. Les classes abstraites sont blanches et les classes concrètes (qui correspondent aux disjoncteurs du modèle précédent) sont colorées [8].

5. Modèle « pour l'interopérabilité ». En 2003, la décision de suivre la norme IEC 61850 [9] ayant été prise, il est nécessaire d'introduire la modélisation des disjoncteurs proposée par la norme et donc de mettre en correspondance les attributs des classes de la norme avec ceux du modèle hiérarchisé précédent. La norme complexifie le modèle, d'une part en ajoutant des informations sur le disjoncteur comme le nom du constructeur et le no de série (abandonnés dans ALCID en 1989) et d'autre part en considérant des équipements secondaires qui participent, avec le disjoncteur, à la réalisation des fonctions de « coupure ».

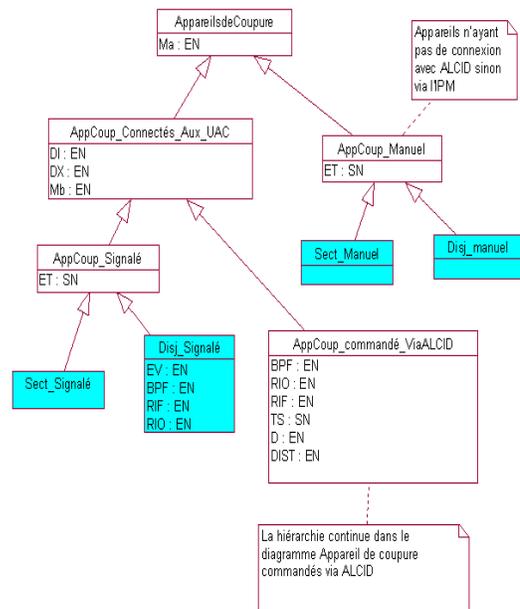


Figure 3. Modèle Hiérarchisé

Ceci transforme le disjoncteur-container du modèle précédent en un « vrai » disjoncteur du point de vue fonctionnel. Cette complexification est indispensable pour permettre à des machines de domaines autres que le contrôle/commande (mais toujours en relation avec la production et le transport de l'énergie) d'interagir avec les IED de contrôle/commande.

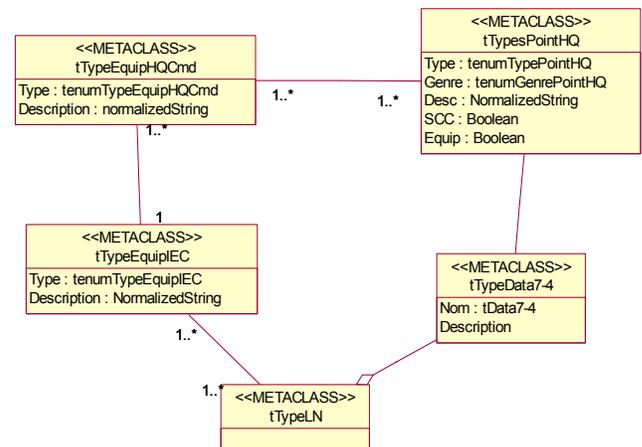


Figure 4. Métamodèle pour la normalisation

6. Métamodèle « pour la normalisation ». En raison de contraintes fixées par Hydro-Québec il ne fallait pas, par exemple, que les ingénieurs responsables du paramétrage soient obligés de connaître la norme IEC 61850. Lors de la conception de la base de données pour les consoles de paramétrage, un métamodèle explicite de normalisation a donc été introduit. Ce métamodèle définit les liens et les contraintes entre les objets indépendamment d'une installation particulière. Le métamodèle permet de fixer les contraintes sur les choix possibles des ingénieurs et

de générer automatiquement des informations concernant l'interopérabilité. La Figure 4 tirée de [10] présente le lien entre les métaclasses qui permettent de générer les points tels que demandés par la norme en partant des points d'entrée-sortie du modèle d'ALCID, bien connus des ingénieurs.

### C. Déclencheurs et difficulté des changements

Nous avons identifié six étapes importantes dans l'évolution de la modélisation des disjoncteurs. Dans cette section, pour chaque étape, nous présentons les déclencheurs et les résistances que nous jugeons les plus importantes et nous les commentons.

#### a) Du modèle « Points d'entrée/sortie » au modèle « universel ».

**Déclencheurs.** Il s'agissait de sortir d'une mentalité orientée RTU (il y a des points qu'il faut envoyer aux centres) afin surtout de faciliter le paramétrage ainsi que l'entrée de nouvelles personnes dans le projet. Par exemple : plutôt que de faire la liste des dizaines de points d'un disjoncteur, il était suffisant de dire qu'il y avait un disjoncteur d'un certain type. Il est aussi important de souligner qu'une partie des individus qui travaillaient sur le projet commençait à être familiers avec l'approche objet.

**Résistances.** Une partie des informaticiens du projet considérait que l'approche objet n'était pas adaptée au contrôle des procédés. Le modèle englobait trop d'éléments hétérogènes : il est clair que mélanger le symbole à afficher sur un poste de travail avec la commande des relais de sortie n'était pas un bon choix.

**Commentaire.** Le modèle trop complexe était une fuite en avant qui s'inscrivait dans une tentative trop ambitieuse d'inscrire le domaine de la commande dans le domaine plus général de l'ingénierie des installations.

#### b) Du modèle « universel » au « container »

**Déclencheurs.** Modèle trop complexe. Échéances de projets trop courtes. Les déclencheurs de 1-2 restent valides.

**Résistances.** Seul celui qui avait proposé le modèle résiste.

**Commentaire.** On peut considérer qu'il y avait eu un passage direct de 1 à 3 et que le modèle 2 n'avait été qu'un modèle pour la démarche (modèle selon l'acception 1 des définitions précédentes de « modèle »). De plus, le temps avait permis à la majorité des intervenants de se familiariser avec l'approche objet.

#### c) Du « container » au modèle « hiérarchisé ».

**Déclencheurs.** Nécessité de mieux comprendre les disjoncteurs et de les catégoriser pour mettre en évidence les éléments communs

**Résistances.** Inutile de faire un nouveau modèle. Tout est bien documenté et le code ne présente pas de gros problèmes.

**Commentaire.** Le modèle explicite était tellement connu par les intervenants qu'ils n'imaginaient pas que quiconque puisse avoir des difficultés à le comprendre.

#### d) Du modèle « hiérarchisé » au modèle « pour l'interopérabilité ».

**Déclencheurs.** Décision de suivre la norme IEC 61850 afin d'introduire une interopérabilité normalisée.

**Résistances.** La norme n'était pas très connue en Amérique, elle était appuyée surtout par les européens. La norme était difficile à comprendre et n'avait pas encore été approuvée.

**Commentaire.** La difficulté à comprendre la norme venait des modèles explicites existants. Il a été très difficile de tout considérer du point de vue de l'interopérabilité surtout en considérant que la norme introduisait beaucoup de nouveaux concepts du domaine des communications.

#### e) Du modèle « pour l'interopérabilité » au métamodèle « pour la normalisation ».

**Déclencheurs.** Besoin de séparer les éléments normalisés de ceux propres à une installation pour faciliter le travail des ingénieurs lors du paramétrage

**Résistances.** Trop de modèles, trop de BD. Difficulté de voir ce qu'est un modèle de normalisation et la différence entre éléments de la norme et éléments de la base de données.

**Commentaire.** Le détail de la norme donne l'illusion que tout est défini.

Dans cette section, nous avons présenté les différentes étapes ayant mené à l'évolution des modèles légués du système ALCID et nous avons illustré ces évolutions dans le cas de la modélisation des disjoncteurs. Nous avons également mis en évidence les déclencheurs ayant conduit à ces évolutions ainsi que les résistances auxquelles il a fallu faire face. Ces différentes évolutions ont été rendues nécessaires pour réaliser la maintenance parfaite du système afin de répondre à des exigences non-fonctionnelles du système.

## IV RESISTANCE AUX CHANGEMENTS

Lorsque dans la littérature l'on traite de systèmes légués, on laisse souvent sous-entendre que les causes principales des coûts élevés de la maintenance sont l'absence de documentation et l'absence des personnes ayant participé au développement. Aucune de ces causes n'était présente dans le cas d'ALCID.

Loin de nous l'idée que la présence de la documentation et des personnes n'aide pas dans la correction des erreurs. Mais, lorsqu'il s'agit de changements importants dus à de nouvelles exigences (l'interopérabilité dans le cas d'ALCID), la documentation et les personnes et, en particulier, les trois modèles légués que nous avons présentés, peuvent être un frein à un développement de qualité. Lorsque les changements des exigences sont accompagnés par des changements importants de la technologie il est même possible que ces modèles empêchent le changement ou que lors du changement on conserve néanmoins le vieux modèle qui, dans le nouveau contexte, devient un frein à la qualité (le modèle *Points d'entrée/sortie* d'ALCID par exemple)

Pour mieux analyser les résistances nous allons préciser ce que nous entendons par « système » : un système est ensemble structuré de composants inter-reliés suivants :

- Matériel et programmes exécutables;
- Listages;
- Documentation avec les modèles explicites et implicites;
- Personnes impliquées. Les personnes peuvent être classées en deux catégories :
  - Ingénieurs du logiciel;
  - Client et/ou utilisateurs.

Dire que lorsque l'on veut modifier un système, le système devient un ensemble de contraintes qu'il faut respecter et qui résistent au changement est une lapalissade. Comme il semble très évident que la résistance au changement est renforcée par un manque de compréhension du système qui est souvent lié à une absence, à une mauvaise ou à trop de documentation.

Nous pouvons considérer qu'ALCID est « bien » documenté et que donc les conditions optimales pour une résistance minimale au changement et pour une « bonne » maintenance corrective sont présentes. À cause de la durée de vie du produit, de nouvelles personnes sont présentes dans le projet et ces personnes ont des connaissances techniques plus « à jour<sup>7</sup> » que les anciens. En raison de leur manque d'expérience, de leur besoin d'imposer leurs idées et de nombreux autres facteurs psychologiques, les nouvelles personnes auront tendance à vouloir casser la résistance du système, à « tout » changer. A travers notre étude de cas, nous mettons en évidence les quatre différents types de résistance que nous avons identifiés :

- la résistance du matériel et du code exécutable;
- la résistance des listages;
- la résistance de la documentation,
- la résistance des humains.

La résistance du matériel et du code exécutable est indépendante des modèles. Plus l'exécutable met en œuvre des fonctions qui permettent de changer en temps réel le fonctionnement de la machine et plus les résistances sont petites. Ce type de résistance est indépendant des modèles et dans ALCID ces résistances ont été minimisées en introduisant un compilateur pour un langage orienté ingénierie des installations. Ce langage permet aux ingénieurs d'écrire des programmes dans un langage graphique proche de leur tâche afin qu'il puissent faire des changements sans recourir aux informaticiens.

La résistance des listages dépend du langage, de l'organisation des fichiers et de la qualité des commentaires et elle est indépendante des modèles. Des normes très strictes ont été respectées dans l'autodocumentation des listages.

Concernant la résistance de la documentation hors des listages, il faut considérer séparément la résistance des modèles implicites de celle des modèles explicites. Pour ces

derniers il est utile de différencier les modèles analytiques des modèles analogiques.

- *Modèles implicites*. S'ils ne sont pas explicités, en tant que modèles, ils aident à structurer les modèles conceptuels dans la tête des personnes. Ce qui a un grand avantage (ils sont plus facilement communicables que lorsqu'ils sont seulement dans le code) et un désavantage potentiel (ils favorisent la « génération » de modèles différents dans la tête de ceux qui avaient développé le logiciel des modèles dans la tête des nouveaux intervenants, ce qui cause souvent bien des conflits). S'ils sont explicités ils changent de catégorie. Mais à ce propos il ne faut pas oublier que l'explicitation a souvent un coût non négligeable même si ce coût est inférieur à celui de l'explicitation en partant des listages. Les modèles implicites en ALCID étaient surtout des modèles conceptuels dans la tête des ingénieurs, modèles pilotés par les caractéristiques des signaux d'entrée et de sortie.
- *Modèles explicites*. Étant donné qu'ils sont « *une représentation simplifiée* » et décrits dans un document, ils acquièrent une objectivité comparable à celle du code. Tout est comme si la réalité était comme le modèle la décrit et il est difficile de changer le modèle surtout si la machine satisfait les besoins pour lesquels elle avait été conçue. Tout changement peut avoir des répercussions multiples sur les listages et dans la vision du problème des intervenants. Ceci peut créer bien plus de problèmes dans un modèle analogique à cause du manque de précision et de formalisme qui le caractérise. Tout au long du développement d'ALCID, nous avons pu constater les nombreuses résistances des modèles explicites. Parmi ces résistances, celles qui nous semblent les plus significatives, pour notre propos, sont celles existant entre le modèle « container » et le modèle « hiérarchisé ». Ce sont les plus significatives parce qu'elles étaient fortes et généralisées même si le changement était loin d'être un grand changement.

La résistance des humains fait appel aux résistances psychologiques des différents intervenants d'un projet. Même s'il est difficile, voire impossible, de séparer les résistances psychologiques dues aux modèles des autres résistances il est important de tenter de faire un peu de lumière. Le modèle en tant que « représentation simplifiée » devient la manière de voir ce qui est représenté et en même temps dans la tête, se « confond » avec ce qui est représenté. Quand il s'agit du modèle du problème (ou du domaine) tout est comme si « le monde » était comme le modèle le décrit. Changer le « monde » devient donc une tâche presque impossible surtout si le modèle est analytique. En effet, la précision du modèle analytique est correcte et ne peut être mise en cause puisqu'elle a permis de mettre en œuvre l'ancienne solution.

<sup>7</sup> Dans une technique jeune comme le génie logiciel, la mise à jour des connaissances techniques est bien plus importante que dans les vieilles techniques ou que dans la science. Ce qui ne veut pas dire que la science et les autres techniques n'évoluent pas.

Dans ALCID on peut considérer que le modèle « point d'entrées/sorties » et le modèle « container » étaient le « monde » : le premier pour les ingénieurs électriques, le deuxième pour les informaticiens néophytes de l'approche objet. Lorsqu'il s'agit d'un modèle qui décrit la conception du logiciel alors les résistances sont moins grandes et surtout moins importantes que d'autres types de résistances (affectives, économiques, etc.).

## V CONCLUSION

Notre conclusion principale est une mise en garde par rapport à l'optimisme généralisé qui porte à croire que les modèles rendent « faciles » la maintenance et la ré-ingénierie des systèmes.

L'expérience d'ALCID nous semble indiquer que parfois les modèles sont un obstacle additionnel à l'évolution des systèmes surtout lorsque les changements ne sont pas des changements fonctionnels mais des changements d'exigences non fonctionnelles (l'interopérabilité dans notre cas). Si, par exemple, il n'avait pas existé pas de modèles explicites avant la décision d'adopter la norme IEC 61850 pour l'interopérabilité, le passage vers un nouveau système aurait été sans doute plus facile. Nous sommes par contre sûrs que les modèles sont fondamentaux pour la maintenance corrective surtout quand elle concerne les exigences fonctionnelles.

Nous croyons aussi que le cas ALCID montre que le « domaine » de la maintenance perfective est un domaine trop vaste et qu'il serait utile de considérer deux types de maintenance perfective: perfective fonctionnelle et perfective non fonctionnelle, car les deux influencent différemment la perception et l'utilité des modèles. À la limite, nous croyons que la différence entre perfective fonctionnelle et perfective non fonctionnelle est plus significative que celle entre perfective et corrective.

Dans notre cas il faut aussi considérer qu'il y avait deux types d'intervenants, les ingénieurs électriques et les informaticiens, et que l'outillage technique différent des ces deux intervenants a eu un impact certain.

L'expérience acquise à travers les différentes évolutions du système ALCID nous amène à appréhender désormais les modèles existants (modèles légués) comme pouvant être la source de résistances aux changements. Il devient alors primordial de prendre du recul pour examiner ces modèles afin de bien répondre à certaines questions comme par exemple : Quand ces modèles ont-ils été conçus? Par qui? Dans quel but? Représentent-ils la perception de la réalité d'une certaine catégorie d'utilisateurs ou de concepteurs? Répondre à de telles questions amène les participants aux projets d'évolution à repositionner les modèles existants dans leur contexte et à les caractériser en fonction de ce contexte.

## REFERENCES

- [1] Swebok : Guide to the Software Engineering Body of Knowledge; www.swebok.org
- [2] IEEE Std 610.12 1990, IEEE Standard Glossary of Software Engineering Terminology
- [3] Ivan Maffezzini, *Prolégomènes à une critique du génie logiciel : partie II*, Génie Logiciel, Septembre 2003.
- [4] Bran Salic, *The pragmatics of Model-Driven Development*, IEEE Software, Vol 20, Issue 5, 2003
- [5] Ed Seidewitz, *What Models Mean*, IEEE Software, Vol 20, Issue 5, 2003
- [6] Michael Jackson, *Problem Frames*, Addison Wesley, 2001.
- [7] Hydro-Québec : Spécification fonctionnelle système - Projet ALCID – HA\_SFS001\_05.0\_880502, mai 1988.
- [8] Hydro-Québec : Système ALCID : Réseau de communication d'installation SEF 0952, juillet 2003.
- [9] ISO/CEI 61850 : Réseaux et systèmes de communication dans les postes; Partie 4:Gestion du système et gestion de projet.
- [10] Hydro-Québec : Spécification BD de normalisation , août 2004.

## ANNEXE 1 : DÉFINITIONS<sup>8</sup>

*Appareil* : Synonyme d'équipement.

*Appareil télécommandable* : Appareil dont il est possible changer l'état à partir d'un poste de travail distant.

*Appareil télésignalé* : Appareil dont il est possible connaître l'état dans un poste de travail éloigné mais qui ne peut pas être commandé distant.

*Disjoncteur* : Équipement primaire capable de couper des courants de court circuit et caractérisé par quatre états : *Ouvert* (pas de passage de courant), *Fermé* (pratiquement pas de résistance au passage du courant), *État intermédiaire* (en train de passer de l'état Ouvert à Fermé ou vice versa), *Défaut*.

*Équipement* : Tout élément physique (ou assemblage d'éléments physiques) responsable d'exécuter des fonctions. On considère deux types de dispositifs : les équipements *primaires* (disjoncteurs, transformateurs, etc.) et les équipements *secondaires* parmi lesquels on retrouve les IED.

*Intelligent Electronic Device (IED)* : équipement spécialisé muni d'une capacité de traitement numérique et connecté avec des liens de communication série à d'autres IED ou à des ordinateurs "clients".

*Interopérabilité* : Attribut qui indique la possibilité de substituer un IED avec un IED d'un autre fournisseur sans que

<sup>8</sup> Toutes nos définitions sont orientées vers une audience qui n'est pas nécessairement experte du domaine du contrôle/commande des installations électriques.

le fonctionnement du système ne change. Dans le cadre d'ALCID l'interopérabilité est réalisée en imposant les classes définies dans la norme IEC 61850.

*Monopolaire* : se dit d'un disjoncteur avec un seul pôle et qui dans un système triphasé ne coupe qu'une phase.

*RTU (Remote Terminal Unit)* : un IED connecté via des cartes d'entrée et de sorties aux câbles que transportent les signaux en provenance de l'installation et doté d'une ou plusieurs lignes de communication vers des centres.. Pour la RTU l'installation n'est qu'un ensemble de points d'entrées et de sorties (analogiques ou numériques).

*Station terminale*. Voir RTU.

*Tripolaire* : se dit d'un disjoncteur avec trois pôles et qui dans un système triphasé coupe le trois phase.