Génie des exigences : impacts de la standardisation de l'interopérabilité pour un système fonctionnellement stable

4ème Conférence Annuelle d'Ingénierie Système
« Efficacité des entreprises et satisfaction des clients »
Centre de Congrès Pierre Baudis, TOULOUSE, 2-4 mai 2006

Ivan Maffezzini
Université du Québec à Montréal
CP 8888 Succ. Centre Ville
Montréal H3C 3P8 Canada
Maffezzini.Ivan@UQAM.ca

Pierre Martin
Hydro-Québec
800, boul. de Maisonneuve est,
Montréal H2L 4M8 Canada
Martin.Pierre.L@hydro.qc.ca

Van Thich Nguyen
Hydro-Québec
800, boul. de Maisonneuve est,
Montréal H2L 4M8 Canada
Nguyen.Van_thich@hydro.qc.c

Résumé. Dans cet article, nous allons considérer les impacts sur le génie des exigences du changement d'une seule exigence de qualité (interopérabilité). Pour diminuer les coûts de l'entretien du logiciel d'un système de contrôle/commande, l'interopérabilité, telle que définie dans la norme CEI 61850, a été mise au centre du changement. Ce choix a eu des impacts non négligeables sur le processus et sur les artefacts du génie des exigences. Changement surtout dans l'élicitation des exigences (lectures et *brain stormings)*, dans l'analyse (nécessité de modéliser le vieux système) et dans la validation (presque impossible). Nous enchaînons sur les enseignements que nous avons tirés de l'introduction d'un changement de qualité s'appuyant sur une norme très complexe dans un système fonctionnellement stable et mûr. Une question sur la place de l'ingénierie dans le domaine des exigences clôt l'article.

Mots clefs : Génie des exigences, Génie logiciel, interopérabilité, processus, temps réel, contrôle/commande, CEI 61850.

Introduction

Il n'est sans doute pas exagéré d'affirmer que les objets techniques ne peuvent pas survivre sans standardisation. En particulier les objets techniques complexes que l'on appelle « systèmes ». Mais s'il est vrai que la standardisation est une nécessité et un « facilitateur » pour la construction des systèmes, il est selon nous évident que, parfois, les standards — surtout lorsqu'ils s'appliquent à des attributs de qualité — peuvent être très compliqués à comprendre, à appliquer et surtout à intégrer dans un processus d'ingénierie des exigences.

Dans cette communication nous adoptons comme définition du terme « exigence » l'acception B) de (IEEE 1233) : « *une condition à atteindre ou une capacité à posséder par un système ou un composant d'un système pour satisfaire un contrat, un standard, une spécification ou d'autres documents officiellement imposés.* » En particulier, nous ne considérerons que l'exigence comme « […] *une capacité* […] *pour satisfaire* […] *un standard* », et ceci non pas parce que « contrat », « spécifications » et « autres documents » ne sont pas importants, mais parce que, dans le projet dont nous allons parler, « standard » véhicule la seule exigence nouvelle par rapport à l'ancien système.

Le choix de l'acception B) est un choix raisonnable car, d'une part, un standard, quand il est la seule exigence nouvelle, transforme l'acception A)[1] en B) et de l'autre nous ne considérerons pas la problématique associée au choix du standard.

1. Contexte

Hydro-Québec (HQ) est la société d'État responsable de la production, du transport et de la distribution de l'énergie électrique au Québec. À partir de 1989, HQ a commencé à introduire dans ses postes de transport et de répartition et dans ses centrales un système de contrôle/commande (ALCID) distribué sur un réseau local Ethernet avec une couche transport conforme à la norme ISO 8073 et un protocole d'application maison.

Le logiciel d'ALCID a été défini, conçu et mis en œuvre à l'interne. Les exigences pour le matériel ont été spécifiées par HQ et la construction a été réalisée par une compagnie externe.

En 1999, HQ a lancé un projet de modernisation (GTPNA 1999) qui aurait dû donner naissance à un nouveau système vers 2006-2007, à la fin du cycle de vie prévu pour ALCID. Bien qu'aucun changement fonctionnel majeur n'était prévu, le nouveau système devait par contre répondre à des architectures très variables, privilégiant ainsi la modularité matérielle et logicielle. L'exigence de pérennité matérielle était aussi au centre des préoccupations,

[1] « *Une condition ou une capacité dont un utilisateur a besoin pour résoudre un problème ou pour atteindre un objectif.* » (IEEE 1233)

Pour améliorer la maintenabilité et, en particulier, les coûts de la maintenance perfective, il a été décidé de centrer la réalisation du nouveau système autour de l'interopérabilité. Après quelques tentatives de spécifications internes, le projet a été aligné sur la nouvelle norme CEI 61850, car même si certaines de ses parties n'avaient pas encore atteint l'état de norme internationale, la vision cadrait avec celle d'Hydro-Québec.

2. Interopérabilité selon CEI 61850

L'interopérabilité est définie dans la norme (CEI 61850) comme « *L'aptitude de deux ou plusieurs IED (*Intelligent Electronic Device*) du même vendeur ou de vendeurs différents d'échanger des informations et d'employer cette information pour une exécution correcte des fonctions spécifiées.* » La norme CEI-61850 s'applique aux réseaux et aux systèmes de communication des postes de transport et distribution de l'énergie électrique. Son objectif principal est « *obtenir l'interopérabilité entre les IED de constructeurs différents.* » Les IED sont les machines programmables qui échangent des données sur le réseau local et qui sont responsables de la conduite et de la protection des équipements du poste.

La norme est constituée de quatorze parties dont les quatre nommées 7-1 à 7-4 concernent les structures de communication pour le temps réel et la partie 6 concerne les échanges entre les consoles de paramétrage. Le respect de ces deux parties assure que tous les éléments sont présents pour permettre aux intégrateurs du système de configurer les IED de constructeurs différents. En effet, des IED qui interopèrent mais dont les consoles ne permettent pas d'échanger les paramètres sont, à toute fin pratique, inutilisables.
Il est important de souligner qu'il ne s'agit pas d'interopérabilité au niveau de la couche application du modèle OSI mais au niveau des processus d'application — la couche application choisi pour la première version de lCEI 61850 est la norme ISO/IEC 9506 connue surtout sous l'acronyme MMS (*Manufactuting Message Specification*) et intitulée *Spécification de messagerie industrielle* en français.

L'interopérabilité est obtenue en permettant la distribution parmi les IED de sous-fonctions, appelées *nœuds logiques*, définies formellement par composition d'éléments plus primitifs. L'interopérabilité est donc liée à la possibilité qu'ont les nœuds logiques d'exécuter des sous-fonctions normalisées qui s'échangent des données dont le nom, la structure et la signification sont fixés par la norme.

À titre d'exemple XCBR est le nœud logique qui représente fonctionnellement les disjoncteurs et permet :

- de les commander ;
- de lire la position ;
- de lire des informations sur leurs conditions de fonctionnement.

Pour pouvoir définir formellement les fonctions d'un poste, la norme introduit quatre-vingt douze (92) nœuds logiques obtenus par composition de vingt-neuf (29) classes de données communes et définit de l'ordre de sept cent (700) données simples. Ces chiffres, plus le nombre de pages (à peu près mille), donnent une idée de la complexité de la norme.

3. Impacts sur le génie des exigences

Dans cette section, nous analyserons le génie des exigences à l'intérieur du cadre défini dans (Maffezzini 2005), en nous appuyant sur les activités explicitement définies dans SWEBOK et, implicitement, dans le guide pour les principes d'opérations (IEEE Std, 1362) et le guide pour la spécification des exigences du système (IEEE Std, 1233).
En particulier, nous analyserons les impacts sur les activités et les livrables dans le cas où le seul changement majeur requis est le respect d'une norme qui définit une caractéristique de qualité. Dans la première partie (3.1), nous analyserons les impacts de la présence de la norme sur le processus et, dans la deuxième (3.2), les impacts sur les artefacts.

3.1 Impacts sur les activités

3.1.1, **Élicitation des exigences**. L'élicitation étant la tâche qui consiste à découvrir les exigences, en particulier en interagissant avec le client-utilisateur, on pourrait penser que cette activité n'est d'aucune importance dans notre contexte, puisque :

1. il n'y a pas de changements fonctionnels ;
2. la seule exigence non fonctionnelle qui change est l'intéropérabilité qui est détaillée de manière très précise en CEI 61850.

En effet, la seule source de nouvelles exigences est la norme[2]. Il s'agit donc d'acquérir les concepts du domaine tel que définis dans la norme. Cette acquisition n'a pas été facile à cause de la complexité et du volume que l'on a signalés dans la section précédente. C'est à cause de cette complexité que l'acquisition des exigences s'est faite en

[2] Ce qui, pour être précis, à cause des connotations psychologiques de « élicitation », devrait nous faire changer ce dernier terme pour « acquisition » des exigences.

parallèle avec la modélisation dans l'analyse des exigences (voir la section suivante).
À cause de la contrainte principale du projet « *pour les utilisateurs il ne doit pas y avoir de différences significatives avec l'ancien système* » qui implique que tous les concepts propres à la norme doivent être cachés, il a fallu faire ressortir *les exigences fonctionnelles induites* pour pouvoir les « cacher ». Cette nécessité de cacher les changements aux utilisateurs et aux techniciens est assez paradoxale si on considère que dans cette phase du génie des exigences il faudrait surtout « faire parler les utilisateurs ».
En ce qui concerne le paramétrage les exigences ont été « extraites » en faisant « comme si' » on pouvait tout montrer aux techniciens pour ensuite considérer comment cacher tout ce qui avait été induit par la norme (le concept de nœud logique, par exemple). L'« extraction » a été réalisée en s'inspirant de la console de paramétrage de ALCID.

3.1.2 Analyse des exigences.
Pour l'analyse des exigences nous considérerons dans l'ordre : *modélisation conceptuelle*, *Classification des exigences*, *conception de l'architecture*, *répartition des exigences* et *négociation des impacts*.

Modélisation conceptuelle.
La modélisation a été réalisée par itération sur le quatre (4) phases suivantes :

1. modélisation en UML (*Unified Modeling language)* des concepts de la norme. Pour cette phase, seules les quatre sections de la partie 7 de la norme ont été considérées.
2. Modélisation en UML des concepts qui ont piloté la mise en œuvre d'ALCID. Pour cette phase la description des équipements d'ALCID a été employée. Lors de cette modélisation, les anciens modèles ont parfois été un frein pour la nouvelle modélisation (RIV 2005)
3. Mise en correspondances des données de la norme avec celles de ALCID. Pour améliorer la qualité du travail deux mises en correspondance en parallèle et avec très peu d'échange d'information ont été préparées. Le troisième intervenant du projet a joué le rôle d'arbitre.
4. Création d'un site WEB avec les modèles UML par un des intervenants pour faciliter le *brain-storming*.

Il est important de souligner que la modélisation conceptuelle a été un puissant élément de formation.
Pour la partie paramétrage un effort particulier a été réalisé pour décrire dans la notation de description de la configuration (définie avec des schéma XML) des cas types. Cela a permis aussi de valider les modèles UML pour le temps réel.

Classification des exigences.
Même s'il n'y avait pas de changements fonctionnels majeurs par rapport à l'ancien système, les exigences ont été reprises dans le but de les classifier et de les mettre, là où il était possible, en correspondance avec celles de la norme.
Les exigences ont été classées en quatre catégories :

1. *Exigences fonctionnelles*. Les dix-huit (18) exigences fonctionnelles ont été, à leur tour, divisées en trois groupes. Là où il a été possible les exigences fonctionnelles ont été mises en correspondance avec la partie 5 de la norme.
2. *Contraintes*. Les contraintes ont été regroupées en trois groupes : matérielles, de mise en œuvre du logiciel et de communication. Une emphase particulière a été mise sur la partie communication en fixant six (6) contraintes sur les piles des protocoles.
3. *Attributs*. Il s'agit des attributs de qualité ou exigences non fonctionnelles de qualité. Seules les performances ont été décrites en détail. Même si le cadre de la norme pour les métriques de qualité ISO 9126 a été retenu comme référence, aucune métrique n'a été imposée. Six (6) métriques sont données aux constructeurs à titre d'exemple parce que « faciles à introduire et à gérer ».
4. *Exigences architecturales*. Les exigences architecturales ont été organisées selon trois vues : logique, matérielle et logicielle.

Conception de l'architecture
Cette activité était absente car il n'y a eu aucun changement par rapport à l'ancien système qui respectait déjà toutes les exigences induites par l'adoption de la norme.

Répartition des exigences
La répartition des exigences entre matériel et logiciel n'a pas été abordée car, pourvu que les IED se comportent comme un agrégat de nœuds logiques normalisés, les constructeurs ont la liberté de repartir les exigences comme bon leur semble s'ils respectent les exigence non fonctionnelles.

Négociation des impacts

Aucune négociation n'a eu lieu car les différents utilisateurs ne sont pas censés connaître les changements.

3.1.3 Spécification des exigences.
Il s'agit de produire les artefacts présentés en 3.2.

3.1.4 Validation des exigences.
Il n'y a pas eu de validation formelle car :

1. À l'interne n'y avait pas d'autres intervenants ayant une connaissance suffisante de la norme ;
2. des intervenants externes avec une bonne connaissance de la norme n'ont pas été trouvés.

Ceci s'explique sans doute par le fait que la norme était à ses débuts.
L'absence d'une validation formelle n'implique pas qu'il n'y ait pas eu de validation. La validation a été faite surtout dans la phase de modélisation (en travaillant de manière séparée et en introduisant un rôle d'« arbitre », par exemple).

3.2 Impacts sur les artefacts
Les artefacts concernant les exigences dans l'ancien système étaient inspirés des normes IEEE :

- études ponctuelles ;
- principes d'opération ;
- spécification des exigences du système ;
- spécification des exigences logicielles.

Dans ce projet plusieurs études ont été réalisées surtout pour s'assurer de la faisabilité de certaines parties. Mais un changement majeur a été introduit : l'artefact central autour duquel se déroule le projet n'a pas été, comme précédemment, la spécification du système mais le document qui contient la modélisation du domaine.
La modélisation du domaine non seulement a permis de synchroniser les exigences spécifiques d'HQ avec la norme mais a aussi permis de revoir les descriptions des équipements (les objets les plus importants du domaine) de l'ancien système.
Aucune spécification des exigences logicielles n'a encore été produite, par contre la conception de la base de données est en cours. Cette approche, pas du tout orthodoxe, est dictée par le besoin de fixer tous les concepts du domaine, et non seulement ceux de la norme, de manière formelle pour faciliter la spécification et la mise en œuvre de fonctions.
Nous avons choisi une approche inspirée de celle de MCEF (*Most Critical Element First*) que nous avons appelé EPSP (Éléments les Plus Stables en Premier) (Maffezzini 2005). Approche qui rompt avec le cycle de vie en cascade tout en gardant une solidité à laquelle les approches agiles renoncent un peu trop facilement.

4. Leçons apprises

Ce travail a été réalisés par un ingénieur du logiciel et deux ingénieurs en électricité, c'est-à-dire un expert des systèmes informatiques et deux experts du domaine.
Le constat le plus important que nous pouvons tirer de ces six ans de travail est le suivant : **en aucun moment les connaissances en informatique ont été d'une importance quelconque**. Ce qui nous permet de tirer une conclusion qui est loin d'être anodine : des notations comme UML et XML sont facilement assimilable par du personnel technique sans formation en informatique quand ils sont employés comme des outils pour définir et analyser les concepts du domaine. Définition et analyse des concepts qui, tout en étant une des conditions *sine qua non* de l'automatisation, ne concernent en rien la technique informatique. Ce qui en bref peut être exprimé ainsi : **le génie des exigences est un domaine complètement séparé du génie logiciel[3].**
À cette « leçon », on peut en ajouter quelques autres qui, même si moins importantes, nous semblent avoir un intérêt assez général :

1. lorsqu'on désire changer une seule exigence et que cette exigence est définie dans les moindres détails dans une norme complexe, il n'est pas suffisant d'écrire, dans le document *Principe d'opérations,* « quelques phrases et quelques renvois » dans le chapitre qui traite du nouveau système. Il faut se rendre maître de la norme et pour faire cela l'idéal est de décrire le plus formellement possible les concepts du domaine sans craindre de s'éloigner de la conception du système (de la solution). De façon plus imagée on peut dire que nous avons constaté l'importance de « reculer pour mieux sauter ».
2. Une norme « orientée vers les constructeurs » comme CEI 61850 implique un travail de filtrage pour faire ressortir ce qui intéresse les clients et les intégrateurs. Travail qui est loin d'être simple[4].
3. Le passage d'un ancien système à un nouveau n'est pas nécessairement facilité par l'existence de modèles formels de l'ancien. Comme nous

[3] Pourquoi le mettre si en évidence ? Parce que la majorité des ingénieurs du logiciel pense le contraire.
[4] À moins évidemment d'avoir participer activement à l'élaboration de la norme, ce qui n'est pas notre cas.

l'avons montré en (Maffezzini Kerhervé 2005) les anciens modèles peuvent être un frein à l'amélioration.

5. Conclusion

Dans cette conclusion, nous proposons deux thèmes à approfondir : 1) les méthodes de description des normes internationales ; 2) le conséquence de privilégier l'acception b) d'exigences de IEEE 1233.

5.1 Format des normes

1. Les normalisateurs devraient mieux soigner leurs modèles qui risquent de devenir des contraintes majeures pour les ingénieurs des exigences. Dans la figure 10 de la partie 6 de CEI 61850, par exemple, on considère que tous les concepts dérivent d'un nom (*tNaming*) au lieu de considérer que le nom est un constituant des différents concepts. Erreur sans doute due à des considérations de programmation, là où il est fondamental de considérer surtout la clarté conceptuelle.
2. les normalisateurs devraient être conscients que UML et XML ne sont que des notations. On a souvent l'impression (et pas seulement dans CEI 61850) que la découverte de notations plus ou moins formelles pousse les normalisateurs à surévaluer la syntaxe et, surtout, à employer la généralisation à n'importe quel prix (voir le point précédent).
3. Les normes devraient dépasser l'étape de documents de format Pdf ou équivalent, pour se transformer en « documents bases de données », accompagnés par des applications de recherche, de tri, etc.
4. L'introduction de deux manières de décrire les éléments normalisés (UML et XML comme en CEI 61850) peuvent créer des problèmes de cohérence et de compatibilité dans des produits « normalisés ».

5.2 Les deux acceptions d'« Exigence »

L'approche implicitement favorisée par les normes IEEE de génie logiciel est à notre avis trop centrée sur la solution. Les tables des matières proposées pour les trois artefacts les plus importants concernant les exigences (*Principe d'opération, Spécification des exigences su systèmes, Spécification des exigences du logiciel*) montrent clairement qu'elles sont centrées sur l'acception B) d'*Exigence* de IEEE 1233. Acception que nous transcrivons ici : « *une condition à atteindre ou une capacité à posséder par un système ou un composant d'un système pour satisfaire un contrat, un standard, une spécification ou d'autres documents officiellement imposés.* »

Mais cette acception, si importante pour la conception, est dangereuse dans les activités de génie des exigences s'il n'y a pas eu des artefacts créés auparavant pour satisfaire l'acception A) (« *Une condition ou une capacité dont un utilisateur a besoin pour résoudre un problème ou pour atteindre un objectif. »*). C'est-à-dire des artefacts qui, en gros, décrivent le domaine du problème. Ce type d'artefacts n'est pas important que pour les systèmes complexes qui doivent résoudre des problèmes jamais résolus mais aussi pour des cas comme celui que nous venons de présenter où toute la complexité fonctionnelle (résolution de problème) avait déjà été réglée dans l'ancien système.

Cette approche de IEEE est sans doute une conséquence du fait que l'ingénierie, depuis toujours, privilégie les solutions — la partie « théorique » étant surtout un support.

Pour mettre en évidence que les exigences appartiennent au domaine du problème avant d'appartenir à celui de la solution, nous terminons cet article avec une question qui est loin d'être une simple question de mots : **faut-il renoncer au terme « ingénierie » lorsque l'on parle d'exigences dans le domaine du problème ?**

6. Références

CEI, CEI-61850 Standard series : *Communication Networks and Systems in Substation*s, 20001-2004.

GTPNA 2, *RE-C-99-04: Rapport final de la phase de planification*, Hydro-Québec, juillet 1999.

IEEE, SWEBOK, 2004 version, http://www.swbok.org/

IEEE, IEEE Std. 1233-1998, IEEE Guide for *Developping System Requirements Specification.*

IEEE, IEEE Std. 1362-1998, IEEE *Guide for Information Technology – System Definition – Concept of Operation Document*.

Maffezzini Ivan, Pierre Martin, Van Thich Nguyen, *Impact de l'interopérabilité sur l'ingénierie de système*, ICSSEA, 2004.

Maffezzini Ivan, Louis Martin « Prolégomènes à une critique du génie logiciel - Partie IV : exigences », *Génie Logiciel*, mars 2005.

Maffezzini Ivan, Brigitte Kerhervé, *Modèles légués*, RIV, 2005.

Biographie

Ivan Maffezzini est professeur d'informatique à l'université du Québec à Montréal et consultant en génie logiciel pour Hydro-Québec depuis 1982. En 1997, il a fondé le laboratoire de recherche multidisciplinaire Institut Trempet. Ses principaux champs d'intérêt sont les fondements du génie logiciel, la conception de systèmes en temps réel et la qualité du logiciel.

Pierre Martin est ingénieur à Hydro-Québec.
Pierre Martin est ingénieur à Hydro-Québec. Il est le responsable du projet pour le passage du système de contrôle/commande de poste à CEI 61850. Il est membre de l'équipe canadienne sur le WG10 CEI TC57 (Gestion des systèmes de puissance et échanges d'informations associés). Il est membre de l'IEEE-PES.

Van Thich Nguyen est ingénieur à Hydro-Québec. Il est le responsable de la normalisation pour la commande des postes.