

Simulation des Systèmes d'Information des Smart Grids - Une approche par points de vue

Rachida Seghiri, Frédéric Boulanger, Claire Lecocq, Vincent Godefroy

▶ **To cite this version:**

Rachida Seghiri, Frédéric Boulanger, Claire Lecocq, Vincent Godefroy. Simulation des Systèmes d'Information des Smart Grids - Une approche par points de vue. XXXIIIème Congrès INFORSID, May 2015, Biarritz, France. pp.65-80. hal-01239680

HAL Id: hal-01239680

https://centralesupelec.hal.science/hal-01239680

Submitted on 8 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simulation des Systèmes d'Information des Smart Grids

Une approche par points de vue

Rachida Seghiri [1,2], **Frédéric Boulanger** [2], **Claire Lecocq** [3], **Vincent Godefroy** [1]

1. *EDF R&D*
   Clamart, France
   prénom.nom@edf.fr
2. *LRI, Supélec*
   Gif-sur-Yvettes, France
   prénom.nom@supelec.fr
3. *Institut Mines-Télécom/Télécom SudParis, SAMOVAR CNRS UMR 5157*
   Evry, France
   prénom.nom@telecom-sudparis.fr

RÉSUMÉ. *Nous proposons de simuler les Systèmes d'Information des Smart Grids afin de valider/critiquer les choix de modélisation des experts métier. Nous proposons une approche par points de vue selon deux aspects information et dynamique tout en ajoutant un point de vue intégration pour le maintien de cohérence. Nous mettons à contribution les techniques l'Ingénierie Dirigée par les Modèles, en particulier la transformation de modèle. Enfin, nous éprouvons notre démarche sur un cas métier Smart Grid.*

ABSTRACT. *We propose to simulate the Information Systems of Smart Grids to validate / criticize the modeling choices of domain experts. We propose a viewpoint based approach regarding two aspects : information and dynamics. We add an integration viewpoint to maintain consistency and we use Model Driven Engineering techniques. Finally, we experience our approach on a Smart Grid use case.*

MOTS-CLÉS : *Simulation, Ingénierie Dirigée par les Modèles, Smart Grid, Système d'Information*

KEYWORDS: *Simulation, Model Driven Engineering, Smart Grid, Information System*

1. Introduction

Un Smart Grid est un réseau électrique intelligent permettant d'optimiser la production, la distribution et la consommation d'électricité grâce à l'introduction des technologies de l'information et de la communication sur le réseau électrique [1]. Les Systèmes d'Information (SI) doivent donc intégrer pleinement les Smart Grids pour les superviser et les piloter. Dans ce contexte, nous proposons de simuler ces SI en adoptant différents points de vue (métier, fonctionnel, applicatif) afin de faire valider ces simulations par les experts impliqués dans l'implantation des Smart Grids. Nous identifions principalement trois verrous à lever pour y parvenir :

1. Tout d'abord, le Smart Grid implique plusieurs expertises car il combine le domaine électrotechnique, celui des télécommunications et les SI. Les modèles à simuler doivent respecter deux critères : (1) s'appuyer sur des standards pérennes, (2) être compréhensibles par les différents experts métier. Nous expérimentons les approches issues de l'Ingénierie Dirigée par les Modèles (IDM) pour concevoir des modèles respectant ces critères. En effet, l'IDM est à l'origine de nombreux travaux facilitant la prise en compte de standards comme UML dans les processus de développement (Authosserre-Cavarero *et al.*, 2012). De plus, ces standards sont effectivement adoptés par les experts Smart Grids (Uslar *et al.*, 2012) ;

2. Ensuite, pour les simuler, les modèles construits doivent être exécutables. Par la simulation, les experts peuvent valider/critiquer leurs choix de modélisation au plus tôt dans le processus de développement. Des langages comme BPMN [2] et UML possèdent désormais des sémantiques d'exécution formalisées permettant le développement d'outils de simulation ;

3. Finalement, une approche par points de vue est indispensable à l'appréhension de systèmes complexes (France, Rumpe, 2007) tels que les SI des Smart Grids mais il faut absolument maintenir la cohérence entre les vues du système. (Lahna *et al.*, 2005) propose de décrire des correspondances au niveau du métamodèle UML enrichi. Cette approche peut être généralisée en définissant des liens de cohérence entre les vues indépendamment du langage de modélisation.

Nous proposons donc une démarche intégrative par points de vue pour la simulation des SI des Smart Grids en profitant des techniques issues de l'IDM comme la transformation de modèle et en adoptant des modèles exécutables. La deuxième section de cet article décrit le contexte industriel dans lequel s'inscrivent nos travaux. La section 3 présente un état de l'art des concepts et des techniques de l'IDM, des framework d'architecture SI et de la simulation de modèles pour le SI. Notre démarche, ainsi que son application à un cas métier, sont présentées dans la section 4. La dernière section conclut cet article et donne nos perspectives de recherche.

---

1. www.smartgrids-cre.fr
2. Business Process Management Notation

2. Contexte industriel

Dans cette partie, nous présentons la problématique industrielle dans laquelle s'inscrivent nos travaux avant d'introduire le cas métier sur lequel nous souhaitons éprouver notre démarche.

2.1. Problématique industrielle

L'équilibre entre la consommation et la production est essentiel au bon fonctionnement des réseaux électriques. Les Smart Grids sont d'autant plus incontournables pour préserver cet équilibre avec l'arrivée massive des véhicules électriques dont la recharge est imprévisible et mobile, ainsi que l'implantation massive des énergies intermittentes (produites par exemple par les éoliennes, les panneaux solaires). En effet, les Smart Grids assurent une gestion automatique et en temps réel des réseaux électriques grâce à de multiples capteurs et points de contrôle commandés à distance.
Les Smart Grids apportent, par essence, de profonds changements au niveau des Systèmes d'Information qui les pilotent : nouveaux flux d'information provenant du réseau électrique, entrée en jeu de nouveaux acteurs tels que les producteurs décentralisés (éolien, photovoltaïque), nouveaux équipements communicants comme le compteur Linky [3], nécessaire conformité aux nouvelles réglementations et directives européennes [4], nouveaux usages (véhicule électrique, maison connectée).
Afin d'appréhender ces paradigmes naissants, il est indispensable d'élaborer des scénarios métier nécessitant d'être éprouvés et validés avant leur adoption finale. Plusieurs démonstrateurs physiques ont été déployés sur le terrain [5]. Ces projets pilotes permettent de mener des expérimentations en conditions réelles pour tester des fonctions et des services comme par exemple le démonstrateur InfiniDrive [6] pour le pilotage des infrastructures de recharge de véhicules électriques ou le démonstrateur Venteea [7] pour l'intégration de forte capacité éolienne dans un réseau rural. Cependant, les démonstrateurs nécessitent que le gestionnaire de réseau de distribution recrute des clients industriels et/ou domestiques qui acceptent d'avoir du matériel à tester chez eux. De plus, leur exploitation reste limitée par les réglementations en cours. Enfin, leur mise en place se révèle souvent longue et coûteuse.
En plus de ces démonstrateurs, des réseaux de distribution d'expérimentation grandeur nature comme Concept Grid [8], implanté à EDF Lab, permettent de tester les nouveaux équipements avant leur installation sur les réseaux du distributeur ERDF [9]. Ces réseaux ont l'avantage de permettre la conduite de stress tests en conditions perturbées, impossibles à réaliser dans le cadre de démonstrateurs, ceux-ci impliquant de vrais

3. www.erdf.fr/Linky
4. www.horizon2020.gouv.fr
5. www.erdf.fr/Carte_demonstrateurs_Smart_Grids
6. avem.fr/actualite-erdf-et-le-groupe-la-poste-lancent-le-projet-infini-drive-a-nice-3450.html
7. www.venteea.fr
8. chercheurs.edf.com
9. Électricité Réseau Distribution France

clients. Cependant, la taille réduite de ces réseaux reste limitante.
Pour palier toutes ces limitations, une troisième voie est la simulation. Cette simulation intègre les trois domaines qui composent les Smart Grids : l'infrastructure électrique (transformateurs, lignes, charges, sources), l'infrastructure télécom (réseaux mobiles, CPL [10]) et enfin les SI qui les pilotent. Des simulateurs spécialisés dans la simulation de réseaux électriques (EMTP-RV, Dymola, PowerFactory, Eurostag, etc.) ainsi que des simulateurs télécoms (OPNET, NS-3, OMNeT ++, etc.) ont déjà validé l'apport de la simulation dans leurs domaines respectifs. Toutefois, la couche SI est souvent reléguée à de simples modèles de calcul de consigne souvent développés en Matlab ou en C++ (Palensky *et al.*, 2014). Nous souhaitons donc simuler aussi le domaine SI et faire valider les simulations obtenues par les experts impliqués dans le déploiement des Smart Grids.

2.2. Cas métier : gestion d'une flotte de véhicules électriques d'entreprise

De nouveaux usages de consommation d'électricité émergent dont, notamment, la mobilité électrique. En France, les pouvoirs publics ont annoncé leur ambition de miser sur deux millions de véhicules électriques en circulation en 2020. Ce parti pris découle de l'objectif, dit du « facteur 4 », visant à diviser par quatre les émissions de gaz à effet de serre à l'horizon 2050 par rapport à son niveau de 1990.
Dans ce contexte, plusieurs projets R&D visent à gérer au mieux l'impact des véhicules électriques sur les réseaux de distribution d'électricité. En effet, la recharge complète d'un véhicule électrique ayant 150 km d'autonomie est équivalente en terme d'appel de puissance à :

- un chauffe-eau si la recharge s'effectue en 8h (recharge normale) ;
- un immeuble si la recharge s'effectue en 1 h (recharge accélérée) ;
- un quartier urbain si la recharge s'effectue en 3 min (recharge rapide).

Les moyens sollicités pour maintenir l'équilibre offre/demande et garantir la qualité de l'électricité fournie pendant les pics de consommation sont coûteux et émetteurs de CO_2 (démarrage de centrales de charbon et de fioul). Des expérimentations en cours testent de nouvelles méthodes pour éviter les contraintes réseau. Par exemple, la Poste et ERDF ont mis en place le démonstrateur InfiniDrive visant à concevoir un système de recharge optimal pour une flotte de véhicules électriques d'entreprise. Ce système propose de piloter la recharge en minimisant la concentration de recharge rapide et accélérée sur les heures de pointe (risque de mise en contrainte du réseau voire du parc de production et de panne sur le réseau), tout en prenant en compte les incitations tarifaires favorisant la recharge en heure creuse.
Par ailleurs, la mobilité électrique implique un changement de paradigme pour le gestionnaire de flotte de l'entreprise. D'une part, le véhicule électrique est limité par son autonomie et ne peut donc pas effectuer n'importe quelle tournée. D'autre part, la recharge d'un véhicule électrique ne se fait pas comme le plein d'essence d'un véhicule

10. Courant Porteur en Ligne

thermique (temps de recharge, disponibilité des bornes).
Dès lors, le processus d'affectation de véhicules aux tournées des agents, la gestion de la flotte de véhicules dans son ensemble et donc le SI qui l'implante sont fortement impactés par l'arrivée massive des véhicules électriques. Nous proposons de modéliser les SI impactés et de les simuler. Cette simulation permettra de valider/critiquer les choix de modélisation et d'anticiper l'éventuel dimensionnement de la flotte. Par exemple, si une forte proportion des tournées implique une distance effectuée supérieure à l'autonomie des véhicules électriques sans possibilité de recharge en cours de route (pas de borne à disposition au cours de la tournée), la simulation permet de trouver la proportion de véhicules thermiques à garder a minima dans une flotte. L'affectation doit aussi privilégier l'utilisation des véhicules électriques car la rentabilité d'un parc de véhicules électriques est proportionnelle au nombre de kilomètres effectués par ces véhicules.
Ce cas métier fait intervenir aussi bien un SI centralisé (celui du pilotage de la flotte de véhicules et du calcul du programme de recharge optimisé), qu'un SI réparti (celui embarqué dans les bornes qui exécutent le programme de recharge calculé en identifiant le véhicule électrique concerné).

3. État de l'art

Les Smart Grids sont, par leur nature même, des systèmes complexes (Monti, Ponci, 2010). L'IDM a démontré à travers de nombreux travaux sa capacité à adresser de tels systèmes (France, Rumpe, 2007). La première partie de cet état de l'art traite de l'IDM. De plus, l'objet de nos travaux est de simuler le domaine SI des Smart Grids. La partie 3.2 présente ainsi les principaux points de vue adoptés dans l'architecture des SI et la partie 3.3 précise les objectifs de la simulation SI et les moyens d'y parvenir.

3.1. IDM

L'IDM prône le paradigme du « tout est modèle » (Bézivin, 2005). En effet, c'est une approche orientée modèle qui couvre tout le cycle de vie d'un système logiciel en utilisant des modèles « productifs » (Bézivin *et al.*, 2004), c'est-à-dire, des modèles exploitables informatiquement (simulation, validation, vérification, génération de code, etc.). Les modèles sont donc des artefacts de premier ordre dans l'IDM. Bien que la définition du concept de modèle ait soulevé de nombreux débats, une définition communément admise est donnée par (Bézivin, Gerbé, 2001) : « *un modèle est une abstraction d'un système, construite selon le bon point de vue, qui permet de répondre à des questions prédéfinies sur ce système en lieu et place de celui-ci* ». De cette définition découle la première relation fondamentale de l'IDM, nommée RepresentationDe, qui lie le modèle et le système qu'il représente. En outre, l'IDM permet de capturer la logique métier d'un domaine au niveau des métamodèles. En effet, un métamodèle décrit l'ensemble des concepts métier manipulés ainsi que leurs relations. Pour être exploitable, tout modèle utilisé au cours du cycle de vie du système à implanter doit se conformer à cette logique métier et donc au métamodèle. Ceci nous amène à la

deuxième relation fondamentale de l'IDM qui relie un modèle à un métamodèle et qui est la relation « ConformeÀ » (Favre, 2004).
L'intérêt majeur de l'IDM est de produire des modèles dont l'exploitation est automatisable. Ceci est rendu possible grâce à la transformation de modèle. La transformation génère un ou plusieurs modèles cible à partir d'un ou plusieurs modèles source, selon une description établie de la transformation (Mens, Van Gorp, 2006). Comme l'illustre la FIGURE 1, la description de la transformation se fait au niveau des métamodèles. Un moteur de transformation exécute cette description.
La transformation de modèle est au cœur d'une démarche dirigée par les modèles. Il est possible d'y recourir sur tout le cycle de vie d'un système pour par exemple le raffinement, la composition, l'analyse ou encore la simulation de modèle. Ainsi, l'IDM promeut l'utilisation de modèles pour construire des SI évolutifs, s'alignant rapidement avec les besoins métier grâce aux techniques de transformation de modèle et facilitant la capitalisation du savoir métier dans les métamodèles élaborés.
Cependant, la multitude de modèles produits et manipulés doit former un ensemble cohérent et représentatif du système final quel que soit le point de vue adopté. C'est pourquoi le maintien de la cohérence entre modèles et transformation de modèles fait l'objet de recherche active dans une discipline comme l'IDM. (Steel, Jézéquel, 2007) propose de typer les modèles en entrée des transformations en utilisant le Model Typing. Celui-ci permet de contrôler les modèles manipulés par les transformations. Autrement dit, une transformation ne prend en entrée qu'un certain type de modèle. Ainsi, tous les modèles conformes à ce type peuvent être manipulés par la transformation en question. Ceci a, de plus, l'avantage d'augmenter la réutilisabilité des transformations écrites en mettant en évidence les caractéristiques communes des modèles. Cette technique de typage est en outre outillée et intégrée dans le langage de méta-modélisation Kermeta [11] qui est basé sur EMOF [12] dans un environnement Eclipse.

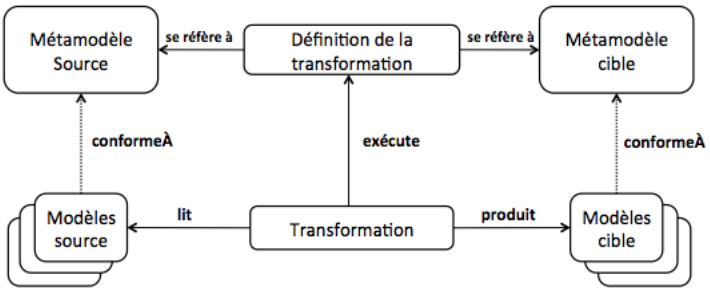


Figure 1. Composants d'une transformation de modèles

11. www.kermeta.org
12. Ecore Meta Object Facility

3.2. Architecture SI

Une définition communément admise du SI est donnée par (Reix *et al.*, 1995) : « *un Système d'Information est un ensemble organisé de ressources : matériel, logiciel, personnel, données, procédures, permettant d'acquérir, traiter, stocker, communiquer des informations (texte, images, sons, etc.) dans et entre des organisations* ». Cette définition du SI a l'avantage de mettre l'accent sur les acteurs impliqués et les informations manipulées outre les moyens techniques mis en œuvre pour les traiter, évitant ainsi de réduire le SI au seul système informatique. Une modélisation appropriée du SI reflète donc les besoins des acteurs (expert du domaine métier, architecte fonctionnel, architecte technique, etc.) en explicitant les informations manipulées.
Étant donnée la complexité des SI actuels, en particulier ceux des Smart Grids, et la multitude des acteurs concernés, une vision monolithique est inappropriée à la construction de SI évolutifs et adaptés aux différents acteurs.
Plusieurs cadres d'architecture adoptent une approche par points de vue. Un point de vue formalise la perspective d'un acteur particulier du SI. Une vue est conforme à ce point de vue. C'est le cas de RM-ODP (Raymond, 1995), de TOGAF [13] [14], de la méthode « 4+1 » de Kruchten (Kruchten, 1995) ainsi que de la norme ISO/IEC/IEEE 42010 [15]. Le SGAM [16] (Uslar *et al.*, 2012) adresse l'architecture du Smart Grid en englobant les trois domaines : SI, réseau électrique et réseau de télécommunication.
En outre, ces frameworks organisent hiérarchiquement les différentes vues en appliquant *« IT follows business »* comme principe : commencer par le point de vue métier et le dériver progressivement jusqu'à l'infrastructure technique déployée en passant par les fonctions et les applications. Souvent, ces cadres d'architecture distinguent quatre points de vue principaux :

Point de vue métier : ce point de vue reflète la vision métier. On y retrouve les objectifs métier de l'entreprise, les processus, ainsi que les acteurs ;

Point de vue fonctionnel : ce point de vue organise le SI en blocs fonctionnels de manière à garantir son évolutivité tout en répondant aux besoins métier de l'entreprise. A l'échelle du SI d'une entreprise, cette structuration devient vite complexe à cause du caractère étendu et transverse des processus métier impactés ;

Point de vue applicatif : ce point de vue structure le SI en blocs applicatifs, chacun implantant un ou plusieurs blocs fonctionnels ;

Point de vue technique : ce point de vue correspond à l'infrastructure technique du SI (matériel informatique et réseaux télécom).

D'une part, ces frameworks proposent une architecture orientée composants (macros processus, blocs fonctionnels, blocs applicatifs, etc.). Les informations sont modélisées soit implicitement et de manière diffuse à l'intérieur des vues (TOGAF), soit

13. The Open Group Architecture Framework
14. www.opengroup.fr/togaf
15. http://www.iso-architecture.org/
16. Smart Grid Archietcture Model

séparément dans une vue dédiée et décorrélée des autres vues (RM-ODP, Kruchten, SGAM).
D'autre part, les modèles utilisés pour les vues — métier, fonctionnel, applicatif — sont souvent des modèles « contemplatifs », destinés à la simple communication ou à la documentation et sont, de ce fait, désincarnés de l'implantation informatique. L'utilisation de modèles exécutables pour la modélisation de ces vues assure une meilleure cohérence des vues du SI et en facilite l'appréhension par les différents acteurs en corrigeant/validant la simulation de ces modèles au plus tôt.

3.3. Simulation du SI

(Shannon, 1975) définit la simulation comme étant « *le processus consistant à modéliser un système réel et à mener des expérimentations sur le modèle obtenu dans le but de comprendre le comportement du système et/ou d'évaluer différentes stratégies concernant son fonctionnement* ». Quel qu'en soit le domaine d'application, la simulation est un moyen d'apprécier les choix des concepteurs sur la structure et le comportement du système modélisé. Elle peut se traduire par l'animation d'un modèle (représentant notre perception du système, qu'il soit existant ou à construire) et l'étude du comportement de ce modèle en fonction des variables en entrée.
La simulation des SI des Smart Grids est d'autant plus cruciale que ces derniers sont en constante et rapide transformation : évolution des cadres législatifs, apparition de nouveaux partenaires, hétérogénéité des interactions avec les clients finaux (compteurs intelligents, web, téléphones, tablettes, etc.). Ainsi, le recours aux modèles exécutables dès les premières phases du cycle de vie des SI des Smart Grids augmente leur évolutivité en apportant une aide supplémentaire à leur validation. La capacité d'exécution et donc de simulation des modèles facilite leur exploration par les experts métier et peut lever les ambiguïtés engendrées par les modèles purement contemplatifs.
L'exécution des modèles est rendue possible par la définition d'une sémantique exécutoire du langage dans lequel ils sont exprimés. La sémantique d'un langage correspond au sens que peuvent prendre les concepts manipulés et leurs agencements lorsqu'ils sont instanciés au niveau des modèles (Jézéquel *et al.*, 2012). La définition de la sémantique du langage dépend de l'objectif poursuivi : simulation, génération de code, vérification, compilation, etc. L'expression de la sémantique d'un langage fait l'objet d'intenses recherches en ingénierie des langages, et en particulier sous la thématique des langages formels (Kleppe, 2007).
Comme en atteste le manifeste d'IBM (Chesbrough, Spohrer, 2006), les axes principaux de l'IDM sont (1) les standards ouverts, (2) l'automatisation et (3) la représentation directe. Compte tenu de ces axes, pour modéliser les différentes vues du SI, nous préconisons l'utilisation de langages standardisés, exécutables et compréhensibles par les acteurs du SI des Smart Grids. Nous identifions plusieurs langages pouvant satisfaire ces critères :

– Un sous-ensemble de UML limité au diagramme de classes et au diagramme d'activité possède désormais une sémantique d'exécution décrite par le standard

fUML [17]. Les diagrammes de classes conviennent pour la description des modèles d'information tandis que les diagrammes d'activité sont adaptés à la description de la dynamique d'un modèle et au comportement attendu des fonctions ;

– Le standard BPMN est un langage de modélisation graphique permettant de décrire tous les aspects d'un processus métier à l'aide d'un seul type de diagramme. Ce formalisme présente l'avantage d'avoir une sémantique d'exécution bien définie permettant le développement d'outils pour la simulation de modèles de processus métier. BPMN est parfaitement adapté à la description de processus métier ;

– Le standard OCL [18], un langage textuel standard d'expression de contraintes, a été ajouté à UML pour exprimer les propriétés difficiles à capturer dans des diagrammes UML. L'exécution d'OCL se fait à travers la transformation de modèle en ciblant un langage d'expression de contrainte de plus bas niveau qui soit exécutable comme MiniZinc (Nethercote *et al.*, 2007), ou à travers son utilisation au niveau du métamodèle avec OCLinEcore [19].

Par la suite, nous présentons la démarche générale que nous proposons pour la modélisation et la simulation des SI des Smart Grid en adoptant une approche par points de vue et en mettant à contribution les techniques issues de l'IDM telles que la transformation de modèle et le Model Typing.

4. Démarche proposée et application au cas métier

Dans cette partie, nous formalisons tout d'abord notre démarche générale avant de la valider sur le cas métier de la gestion d'une flotte de véhicules électriques.

4.1. Démarche générale

Toute simulation d'un système commence par sa modélisation. Pour modéliser des systèmes complexes comme les SI des Smart Grid, nous adoptons une approche par points de vue. Celle-ci facilite la conception des modèles par les acteurs impliqués en séparant leurs préoccupations respectives. Elle permet également de présenter les modèles obtenus, ainsi que les résultats de simulation de manière plus compréhensible à ces acteurs, car chaque point de vue n'utilise que les concepts métier propres à chaque acteur, selon sa perspective sur le SI.

L'approche que nous proposons identifie quatre points de vue selon les acteurs, spécialistes et interlocuteurs intervenant dans la modélisation et la validation par simulation du SI. Cette approche est illustrée FIGURE 2. Les trois premiers points de vue — métier, fonctionnel et applicatif — se retrouvent dans les approches d'architecture d'entreprise existantes. Cependant, ces approches traitent essentiellement l'aspect *dynamique* selon le point de vue abordé. Notre approche intègre ces points de vue selon

17. Foundational UML
18. Object Constraint Language
19. wiki.eclipse.org/OCL/OCLinEcore

deux aspects : *information* et *dynamique*. Nous nous limitons à trois points de vue — métier, fonctionnel et applicatif — dans un premier temps mais souhaitons étendre nos futurs travaux au point de vue technique également. Ainsi nous retrouvons :

1. L'aspect *dynamique* d'un point de vue métier qui reflète la perception des experts métier. On y retrouve les processus métier de l'entreprise, décrits en utilisant les concepts métier, sans référence aux détails d'implémentation. Nous préconisons l'utilisation de formalismes standard pour la modélisation de processus métier qui soient exécutables, tels que les diagrammes d'activité UML ou les diagrammes BPMN dans une perspective de simulation. Des langages spécifiques à un domaine (DSML) peuvent également être utilisés ;

2. L'aspect *dynamique* d'un point de vue fonctionnel qui reflète la perception de l'architecte fonctionnel et décrit les fonctions qui manipulent les données métier. Ces fonctions sont regroupées en blocs. Chaque objet métier identifié dans l'aspect Information de la vue métier correspond à un unique bloc fonctionnel. Ceci garantit la construction de blocs fonctionnels fortement décorrélés, avec une forte cohésion interne. Dans chaque bloc fonctionnel, on retrouve les opérations correspondant à une activité donnée du processus qui impacte l'objet métier impliqué ;

3. L'aspect *dynamique* d'un point de vue applicatif qui reflète la perception de l'architecte applicatif. Il aboutit à l'architecture applicative rassemblant les modules logiciels qui implantent les blocs fonctionnels. Dans un premier temps, il est conseillé de dresser un inventaire de l'existant applicatif et d'en extraire les modules capables de réaliser les opérations des blocs fonctionnels. Ensuite, si aucune application ou module existant ne peut répondre au besoin des nouveaux processus métier, l'architecte technique fait le choix des nouveaux composants applicatifs à mettre en place. En plus d'identifier les composants applicatifs existants ou à développer, l'architecte applicatif spécifie leurs interconnexions (échange de messages, synchronisation de données, transfert de fichiers périodique, etc.).

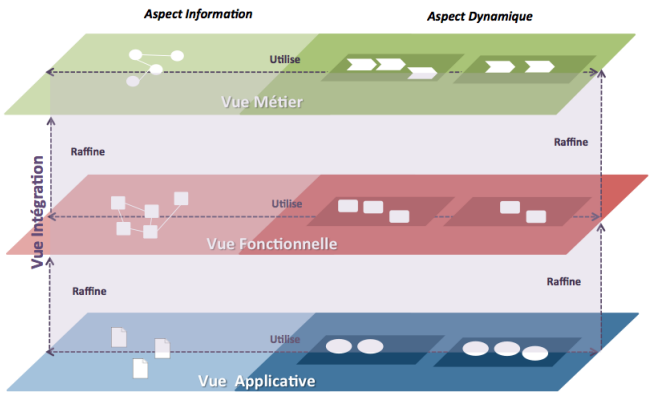


Figure 2. Démarche proposée

Nous proposons d'étendre chacune des trois vues classiques par l'aspect *information*

qui décrit les données qu'utilisent les modèles de l'aspect *dynamique*. L'aspect *information* permet d'avoir un modèle explicite des données utilisées dans chacune des vues métier, fonctionnelle et applicative :

1. L'aspect *information* d'un point de vue métier établit le « modèle de données métier » qui décrit les entités métier majeures manipulées par le processus. Ce modèle est peu sujet au changement, sauf évolution importante des pratiques métier. Il est aussi à l'origine du découpage en bloc par entité métier de la vue fonctionnelle ;

2. L'aspect *information* d'un point de vue fonctionnel établit le « modèle de données fonctionnelles » qui donne une description fine des données utilisées par les blocs fonctionnels nécessaires à la réalisation de processus métier. Elle décrit leurs caractéristiques et leurs relations sous forme de diagrammes de classes par exemple ;

3. L'aspect *information* d'un point de vue applicatif établit le « modèle de données applicatives », qui dépend fortement des applications choisies : il décrit les formats de données compatibles avec les modules applicatifs.

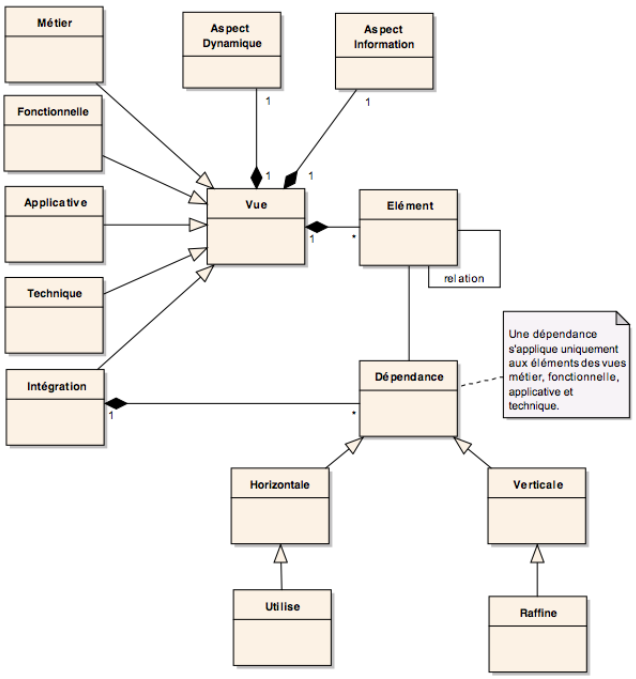


Figure 3. Métamodèle de la démarche proposée

Notre approche se distingue des démarches classiques par l'ajout du **point de vue intégration**. Ce point de vue permet une intégration « verticale » (entre les vues) et « horizontale » (à l'intérieur d'une seule vue). La FIGURE 3 donne un métamodèle de notre démarche en explicitant les concepts abordés et leurs relations. L'intégration verticale décrit les relations entre les trois points de vue précédents, étendus à leur aspect information. Ce point de vue intégration est donc transverse et permet de mo-

déliser explicitement les notions de raffinement à travers la classe « Raffine », tant sur les aspects *dynamique* (processus, fonctions, applications) que sur les aspects *information* (représentation des données métier par des données fonctionnelles, codage des données fonctionnelles dans des formats applicatifs). Par exemple, elle permet de vérifier qu'une application implémente bien tous les blocs fonctionnels nécessaires au déroulement d'un processus métier. Cette vue donne ainsi accès aux informations de traçabilité qui permettent de déterminer l'impact d'une modification ou d'une défaillance d'un module applicatif sur les processus métier. Elle permet aussi de vérifier qu'un format applicatif permet d'encoder les données fonctionnelles qui représentent les données métiers, et de déterminer les éventuelles transformations de modèle nécessaires au déploiement. De la même façon, la vue intégration permet des vérifications « horizontales » à l'intérieur de chacune des vues. La classe « Utilise » assure donc la compatibilité des données échangées entre les tâches d'un processus métier, les fonctions d'un bloc fonctionnel ou entre les modules d'une application.

4.2. Mise en œuvre de la démarche sur le cas métier

Dans cette partie, nous éprouvons notre démarche au cas métier de la gestion d'une flotte de véhicules électriques. Nous construisons les modèles adéquats pour les vues métier, fonctionnelle et applicative en adoptant des langages exécutables. La cohérence est modélisée dans la vue intégration. L'architecture globale du cas métier est illustrée dans la FIGURE 4.

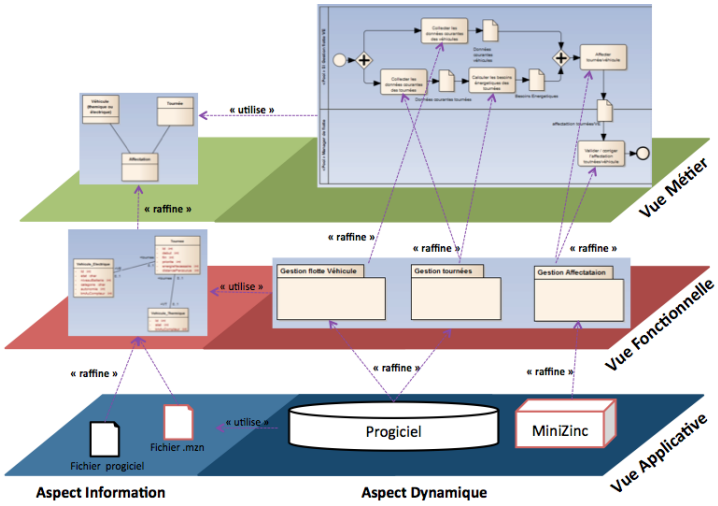


Figure 4. Architecture Globale du SI de gestion de flotte de véhicules électriques selon notre démarche

Vue métier

Pour cette vue nous préconisons l'utilisation de BPMN comme langage de modélisation exécutable. Nous modélisons et simulons le processus métier d'affectation de véhicule (électrique ou thermique) à une tournée avec l'outil Enterprise Architect [20]. Nous utilisons donc un modèle de ce processus métier écrit en BPMN pour décrire l'aspect *dynamique* de la vue métier (FIGURE 4). Il s'agit de collecter les données relatives aux véhicules ainsi qu'aux tournées à effectuer, de calculer l'énergie nécessaire à chaque tournée et l'affectation véhicule/tournée avant de faire valider cette dernière par le manager de flotte. Pour l'aspect *information* de la vue métier, nous modélisons les objets métier sous la forme d'un diagramme de classes UML représentant les concepts de véhicule, de tournée et d'affectation tournée/véhicule. Le choix des langages de modélisation et de l'outil de simulation est motivé par les pratiques du domaine. En effet, la Commission Électrique Internationale a adopté Enterprise Architect comme outil pour maintenir et distribuer le CIM [21](Uslar *et al.*, 2012), un modèle d'information commun pour le domaine électrique [22].

Vue fonctionnelle

Nous identifions trois blocs fonctionnels : un bloc pour la gestion de la flotte de véhicules (électriques et thermiques), un bloc pour la gestion des tournées, un bloc pour la gestion de l'affectation. Ces blocs servent à modéliser l'aspect *dynamique* de la vue fonctionnelle (FIGURE 4). En effet, les blocs fonctionnels sont responsables des traitements nécessaires à la réalisation du processus métier. Par exemple, nous modélisons l'affectation véhicule/tournée sous la forme de contraintes OCL (FIGURE 5) : pour affecter un véhicule à une tournée, il faut que l'énergie nécessaire à celle-ci soit inférieure à l'autonomie de la batterie.

```
context Tournee
inv:    self.VE.autonomie * self.VE.niveauBatterie > self.distanceParcourue

context Tournee
inv:    self.VE <> undefined xor self.VT <> undefined

context Tournee::kmElec(): int
body:   (Tournee::allInstances()->collect(t.VE <> undefined |t.distanceParcourue))->sum()
```

Figure 5. Contraintes OCL pour la gestion d'affectation (vue fonctionnelle)

Nous considérons un premier cas où il n'est pas possible de recharger la batterie au cours de la tournée. L'aspect *information* de la vue fonctionnelle prend la forme de données fonctionnelles modélisées par un diagramme de classes sur lequel s'appliquent les contraintes OCL (FIGURE 4). OCL est adapté aux diagrammes de classes. De plus, OCL est un langage standardisé et exécutable pour l'expression de contraintes. Il est possible de modéliser les autres algorithmes de traitement (calcul des tournées à partir de bon de travaux, calcul de l'énergie nécessaire à une tournée, etc.) à l'aide de diagrammes d'activité exécutables.

---

20. www.sparxsystems.com.au
21. Common Information Model
22. www.sparxsystems.com.au/press/articles/iec.html

Vue applicative

Dans cette vue, nous identifions les applications nécessaires à l'implantation des blocs fonctionnels. Dans notre cas, le patrimoine applicatif de l'entreprise dispose déjà d'applications pour la gestion de tournées (calcul de tournées optimisé à partir de bons de travaux) et la gestion de véhicules (administration, maintenance, etc.). Nous choisissons MiniZinc pour modéliser les contraintes au niveau applicatif (FIGURE 6).

```
constraint forall (i in Tournees) (
%affectation de vehicule electrique si l autonomie l autorise
constraint forall(i in  Tournees, j in VehiculesElec)
(tourneeVehicule[i]= identifiantVE[j] -> (autonomie[j]*niveauBatterie[j] > distanceTournee[i] /\
kmElec[i]=distanceTournee[i]));

% affectation d un vehicule thermique et dans ce cas kmElec est nul
constraint forall(i in  Tournees, k in VehiculesTherm)
(tourneeVehicule[i]= identifiantVT[k] -> kmElec[i]=0);

%maximiser le nombre de km de tounrnee fait par les vehicules electriques
solve maximize sum(i in Tournees)(kmElec[i]);
```

Figure 6. Modélisation des contraintes en MiniZinc (vue applicative)

MiniZinc est un langage de modélisation et de résolution de contraintes de niveau intermédiaire qui a pour vocation de devenir un langage de modélisation standard dans le domaine de la programmation par contraintes. Pour l'aspect *information*, nous retrouvons les fichiers contenant les données nécessaires aux différentes applications (FIGURE 7).

```
nbreVE = 4;
identifiantVE = [1, 2, 3, 4];
nbreVT = 4;
identifiantVT = [11, 22, 33, 44];
autonomie = [1500, 1200, 1500, 1200];
niveauBatterie = [8, 6, 5, 2];
nbreTournees = 4;
distanceTournee = [1000, 700, 800, 1600];
```

Figure 7. Fichier de données MiniZinc (vue applicative)

Vue intégration

Nous avons modélisé les dépendances entre les vues par la relation «raffine» et entre aspects d'une même vue par la relation «utilise». Nous proposons un modèle de ces relations dans la FIGURE 8. Nous envisageons d'implémenter ces dépendances en utilisant le Model Typing pour assurer la cohérence entre ces vues et entre les aspects *information* et *dynamique* au sein d'une même vue. En plus de ces dépendances, la transformation peut jouer le rôle de garant de la cohérence. Par exemple, en transformant les contraintes OCL en contraintes MiniZinc tout en les raffinant par l'ajout de la fonction de maximisation de la distance parcourue par les véhicules électriques. ATL [23] permet d'écrire ce genre de transformation (Chenouard *et al.*, 2010).

23. https://eclipse.org/atl/

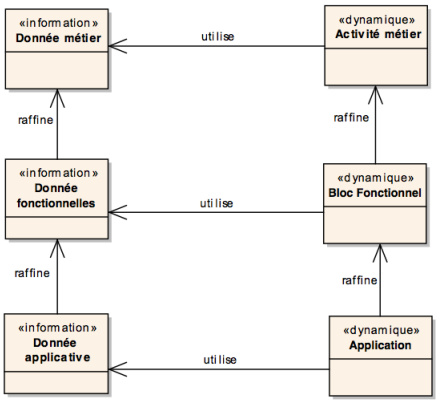


Figure 8. Modèle de la vue intégration

5. Conclusion et perspectives

Dans cet article, nous avons démontré l'importance de la simulation des SI des Smart Grids. Pour y parvenir, nous proposons une approche de modélisation et de simulation par points de vue. Nous retenons les vues métier, fonctionnel et applicatif. Outre l'aspect dynamique de ces vues, nous explicitons la modélisation de l'aspect informationnel. Nous proposons ensuite de rajouter une vue intégrative permettant de maintenir la cohérence horizontale (entre l'aspect dynamique et l'aspect information), ainsi que la cohérence verticale (entre les trois vues traditionnelles). Le Model Typing et les techniques de transformation de modèles sont de bons candidats pour la construction de cette vue. Pour la simulation des modèles, nous proposons l'utilisation de langages de modélisation exécutables en privilégiant les standards tels que fUML, BPMN ou encore OCL.
Nous avons ensuite commencé à valider notre démarche sur un cas métier Smart Grid : la gestion d'une flotte de véhicules incluant des véhicules électriques et en particulier l'affectation quotidienne des véhicules aux tournées.
Nos travaux portent maintenant sur la co-simulation des différentes vues tout en maintenant leur cohérence en mettant en œuvre le Model Typing et les transformations de modèle. Nous projetons aussi de généraliser notre démarche en intégrant la vue technique afin de construire une simulation complète du SI.

Bibliographie

Authosserre-Cavarero A., Bertrand F., Blay-Fornarino M., Collet P., Dubois H., Ducasse S. *et al. (2012). Interopérabilité des systèmes d'information: approches dirigées par les modèles. In* Inforsid 2012*, p. 11–30.*

Bézivin J. (2005). On the unification power of models. Software & Systems Modeling*, vol. 4, n° 2, p. 171–188.*

Bézivin J., Blay M., Bouzhegoub M., Estublier J., Favre J.-M., Gérard S. et al. (2004). Rapport de synthèse de l?as cnrs sur le mda (model driven architecture). *CNRS, novembre*.

Bézivin J., Gerbé O. (2001). Towards a precise definition of the omg/mda framework. In *Automated software engineering, 2001.(ase 2001). proceedings. 16th annual international conference on*, p. 273–280.

Chenouard R., Granvilliers L., Soto R. (2010). Using atl to define advanced and flexible constraint model transformations. *arXiv preprint arXiv:1002.3078*.

Chesbrough H., Spohrer J. (2006). A research manifesto for services science. *Communications of the ACM*, vol. 49, n° 7, p. 35–40.

Favre J.-M. (2004). Towards a basic theory to model model driven engineering. In *3rd workshop in software model engineering, wisme*, p. 262–271.

France R., Rumpe B. (2007). Model-driven development of complex software: A research roadmap. In *2007 future of software engineering*, p. 37–54.

Jézéquel J.-M., Combemale B., Vojtisek D. (2012). *Ingénierie dirigée par les modèles: des concepts à la pratique...* Ellipses.

Kleppe A. G. (2007). A language description is more than a metamodel.

Kruchten P. B. (1995). The 4+ 1 view model of architecture. *Software, IEEE*, vol. 12, n° 6, p. 42–50.

Lahna B., Roudies O., Giraudin J.-P. (2005). Une approche multivue pour la conception de systèmes d'information à base de composants. In *Inforsid*, p. 45–60.

Mens T., Van Gorp P. (2006). A taxonomy of model transformation. *Electronic Notes in Theoretical Computer Science*, vol. 152, p. 125–142.

Monti A., Ponci F. (2010). Power grids of the future: Why smart means complex. In *Complexity in engineering, 2010. compeng'10.*, p. 7–11.

Nethercote N., Stuckey P. J., Becket R., Brand S., Duck G. J., Tack G. (2007). Minizinc: Towards a standard cp modelling language. In *Principles and practice of constraint programming–cp 2007*, p. 529–543. Springer.

Palensky P., Widl E., Elsheikh A. (2014). Simulating cyber-physical energy systems: challenges, tools and methods. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 44, n° 3, p. 318–326.

Raymond K. (1995). Reference model of open distributed processing (rm-odp): Introduction. In *Open distributed processing*, p. 3–14. Springer.

Reix R., Fallery B., Kalika M., Rowe F. (1995). *Systèmes d'information et management des organisations*. Vuibert.

Shannon R. E. (1975). Systems simulation.

Steel J., Jézéquel J.-M. (2007). On model typing. *Software & Systems Modeling*, vol. 6, n° 4, p. 401–413.

Uslar M., Specht M., Dänekas C., Trefke J., Rohjans S., González J. M. *et al. (2012).* Standardization in smart grids: Introduction to it-related methodologies, architectures and standards. *Springer Science & Business Media.*