# Vers un entrepôt de données et des processus : le cas de la mobilité électrique chez EDF

Kevin Royer

# THESE

Pour l'obtention du Grade de
DOCTEUR DE L'ECOLE NATIONALE SUPERIEURE DE MECANIQUE ET D'AEROTECHNIQUE
(Diplôme National – Arrêté du 7 août 2006)

Ecole Doctorale :
S2IM : Sciences et Ingénierie pour l'Information et les Mathématiques

Secteur de Recherche : INFORMATIQUE & APPLICATIONS

Présentée par :

Kevin ROYER

**********

Vers un entrepôt de données et des processus :
le cas de la mobilité électrique chez EDF

**********

Directeur de thèse : Ladjel BELLATRECHE

**********

Soutenue le 30 mars 2015

devant la Commission d'Examen

**********

## JURY

| | | |
|---|---|---|
| Salima BENBERNOU | Professeur, LIPADE, Univ. Paris Descartes | Présidente |
| Chantal REYNAUD | Professeur, LRI, Univ. Paris Sud | Rapporteur |
| Claude GODART | Professeur, LORIA, Univ. de Lorraine | Rapporteur |
| Anne MONCEAUX | Dr. - ingénieur, Airbus Group, Toulouse | Examinatrice |
| Frédérick GARCIA | Dir. de recherche, INRA, Toulouse | Examinateur |
| Ladjel BELLATRECHE | Professeur, LIAS/ISAE-ENSMA | Examinateur |
| François de SOUSA LOPES | Ingénieur, EDF R&D, Moret | Examinateur |

# Remerciements

Je tiens à remercier **Ladjel BELLATRECHE**, mon directeur de thèse, pour son encadrement au cours de cette thèse. Ses conseils et ses idées mais aussi ses encouragements m'ont permis d'accomplir ce travail, et ce faisant d'apprendre beaucoup sur le domaine étudié comme sur moi-même.

Je remercie également toute l'équipe du projet «Véhicules Électriques» ainsi que la direction du département EPI d'EDF R&D de m'avoir accueilli, accompagné et surtout fait confiance pour mener les travaux présentés dans ce manuscrit. Et plus particulier mon encadrant **François de SOUSA LOPES** pour son soutient, sa disponibilité et sa bonne humeur.

Merci à **Chantal REYNAUD** et **Claude GODART** de me faire l'honneur de rapporter ma thèse. Avec eux je souhaite remercier les examinateurs **Salima BENBERNOU**, **Anne MONCEAUX** et **Frédérick GARCIA** d'avoir accepté de faire partie du jury. Merci de l'intérêt que vous portez à mes travaux.

Mes remerciements vont également aux structures qui ont permis de réaliser cette thèse. D'une partie l'ANRT grâce à laquelle le programme CIFRE facilite le partenariat, autour d'une thèse, d'une entreprise et d'un laboratoire de recherche. D'autre part, le LIAS dont j'ai eu la chance de bénéficier des compétences comme celles de Stéphane JEAN, Mickael BARON et Zoé FAGET que je remercie. Et EDF, plus particulièrement le département EPI et le groupe E25, qui m'a apporté les connaissances qu'il me manquait sur la mobilité électrique et son intégration dans les problématiques énergétiques globales.

Merci à mes collègues et/ou ami(e)s, toujours présents, pour l'équilibre qu'ils m'ont permis de trouver : José, Tristan, Émilie, Vivien, Bastien, Bill, Maxime, Thomas, Louise, Juan, Christian, Mathieu, Samuel, Jean-Michel, Baptiste, Pierre-Yves, Houssam , Didier pardon à ceux que j'aurais omis, à vous tous un grand merci ainsi qu'aux basketteurs des Renardières.

Je remercie chaleureusement les doctorants, les anciens comme ceux en cours de thèse, du LIAS pour leur accueil toujours amical et leur aide, merci.

À mes commandants qui m'ont permis de le devenir : Bruno, Patric, Jean-Noel, Erwin, sans oublier le CVVFR et l'Aneg, merci pour tous les moments partagés, puissent-ils y en avoir d'autres.

Je remercie sincèrement mes proches, mes parents, mes grands-parents et mon frère pour m'avoir porté dans cette thèse comme depuis toujours.

Enfin, et surtout, je remercie ma compagne, Amélie, pour m'avoir porté et supporté dans cette entreprise, merci.

*A ceux qui m'ont permis d'être qui je suis.*

# Table des matières

# Partie II Contributions

## Partie III Cas d'étude et conclusions

# Introduction générale

Dans cette partie nous présentons le cadre dans lequel cette thèse s'est déroulée. Celle-ci s'inscrit dans les travaux de recherche et de développement de l'entreprise EDF [1]. Précisément, elle porte sur les méthodes et les outils de construction et d'exploitation d'un système d'information alimenté au fil de l'eau par des données provenant de capteurs placés sur des bornes de recharges dédiées aux véhicules électriques et aux véhicules hybrides rechargeables. Ces travaux sont supervisés par la Direction de la Mobilité Électriques (DME) de l'entreprise EDF dans le cadre d'expérimentations grandeur nature de nouvelles solutions de mobilité électrique individuelles.

## 1 Contexte et besoins

### 1.1 Un domaine en pleine expansion

L'histoire du véhicule électrique ($\mathcal{VE}$) remonte au début du XX$^{e}$ siècle. Le développement de cette technologie s'est poursuivi depuis lors avec des épisodes intenses au moment des chocs pétroliers. Ces derniers entraînant des hausses du prix des carburants, le $\mathcal{VE}$ apparaissait alors comme une alternative économiquement viable. Puis lorsque les prix des carburants redescendaient les projets de création de $\mathcal{VE}$ périclitaient. Cependant, depuis les années 1990, les prix des carburants ne redescendent plus aussi bas. De plus, la perception par les citoyens, les pouvoirs politiques et les entreprises de notions environnementales ont permis de soutenir dans la durée des projets de $\mathcal{VE}$. La corrélation de ces deux facteurs fait qu'aujourd'hui ce domaine est activement soutenu par les industriels (avec de nombreux modèles de $\mathcal{VE}$ disponibles) et le gouvernement (crédits d'impôts [2], financement de projets [3] pour plus d'un milliard d'euros).

1. Électricité De France
2. http://www.developpement-durable.gouv.fr/Plan-de-relance-du-logement-des.html
3. http://www.developpement-durable.gouv.fr/Une-loi-en-faveur-du-developpement.html

Les $\mathcal{VE}$ ont bénéficié de l'évolution technologique et, en retour, y ont contribué. Des entreprises telle que SAFT [4], leader mondial des batteries de haute technologie implanté à Poitiers, ont fourni des efforts remarquables contribuant à ces avancées. Nous pouvons également citer le développement spectaculaire d'outils de positionnement comme le GPS (*Global Positionning System*), de communication comme le GSM (*Global System for Mobile*)et le GPRS (*General Packet Radio Service*), etc.

Cette évolution a transformé le $\mathcal{VE}$ en véritable capteur mobile capable, entre autres, de restituer ses enregistrements de consommations, trajets, etc. L'ensemble de ces données émises par les $\mathcal{VE}$ représente des montagnes de données à disposition des ingénieurs et des experts d'EDF. Leur analyse permet de comprendre les habitudes de recharge des conducteurs, qu'il s'agisse de particuliers ou d'usagers de véhicules de fonction ou d'entreprise. Cette analyse permet d'établir des politiques de facturation adéquates pour la recharge des véhicules et favorise l'émergence de modèles d'affaires efficaces d'un point de vue environnemental et économique.

Pour se positionner dans ce marché en plein essor, EDF a mis en place un projet de recherche et développement (R&D) qui lui est dédié : le projet «Véhicules Électriques». Ce projet a vocation a travaillé sur le domaine dans lequel le $\mathcal{VE}$ s'inscrit, appelé mobilité électrique ($ME$). C'est dans ce contexte que s'inscrit cette thèse CIFRE [5].

## 1.2 Situation globale des $\mathcal{VE}$ : point de vue d'EDF

Pour mieux comprendre les enjeux de cette thèse, nous présentons le contexte du projet pour EDF.

La France est aujourd'hui le deuxième marché européen avec des milliers de nouveaux $\mathcal{VE}$ immatriculés [3] chaque année (voir figure 1). L'entreprise EDF cherche à mettre en place un système d'information fiable, pour l'analyse de données issues de ces véhicules, dans le but d'offrir des services de bonne qualité aux usagers, et de permettre à l'entreprise EDF d'établir un politique de facturation adéquate.

Afin de recueillir un volume significatif d'informations, un nombre important de $\mathcal{VE}$ a été utilisé. Chaque $\mathcal{VE}$ est équipé de capteurs émettant des données vers un système de contrôle et d'acquisition de données *SCADA* [6] (voir figure 2). Ces données sont alors stockées dans des bases de données locales, puis remontées sur les serveurs d'EDF pour y être analysées. Des solutions d'analyse et d'entreposage de données deviennent alors indispensables.

Une autre dimension que la DME doit prendre en compte est la diversité des bornes de recharge. Elles proviennent de différents constructeurs qui développent chacun leurs propres solutions technologiques. Elles peuvent être installées chez un particulier, en voirie ou encore dans des parkings publics ou privés. En conséquence, chaque constructeur offre ses propres

4. `http://www.saftbatteries.com/fr/solutions-du-marche/vehicules-electriques-et-hybrides`
5. Conventions Industrielles de Formation par la Recherche
6. Supervisory Control And Data Acquisition

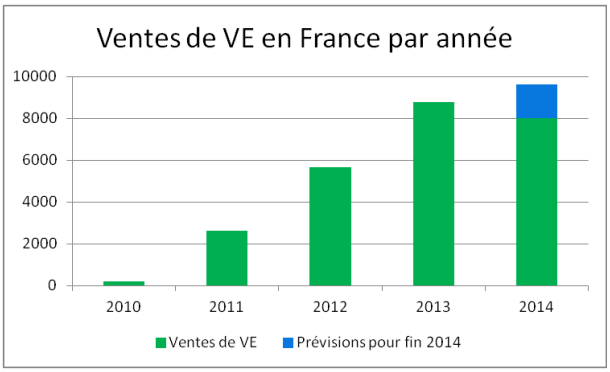


FIGURE 1 – Ventes des VE en France entre 2010 et 2014

formats, unités de mesures, systèmes de nommage, etc.

Pour réduire cette hétérogénéité, le recours aux ontologies de domaine est une solution envisageable. La présence d'une ontologie jouera le rôle d'un schéma global partagé par l'ensemble des sources de données. Contrairement à un modèle conceptuel qui prescrit une base de données selon des besoins applicatifs (orientés application), une ontologie est développée selon une approche descriptive (orientée domaine). En plus de la capacité de conceptualisation, elle permet de décrire les concepts et les propriétés d'un domaine donné indépendamment de tout objectif applicatif et de tout contexte hormis le domaine sur lequel porte l'ontologie. Un autre avantage de l'utilisation des ontologies dans la construction d'un entrepôt de données est la possibilité d'offrir aux utilisateurs finaux la possibilité d'interroger l'entrepôt sans se soucier de son implémentation physique.

Une fois réalisé, cet entrepôt sera exploité par des requêtes issues des processus métiers des ingénieurs d'EDF. L'expérience montre que ces processus sont répétitifs, souvent peu formalisés, et exprimés dans des langages hétérogènes sur les concepts du domaine. Cette situation oblige les ingénieurs et experts d'EDF à se focaliser sur des calculs répétitifs plutôt que sur l'interprétation des résultats obtenus par ces processus. Offrir une fonctionnalité évitant ceci est une tâche difficile car les experts et ingénieurs utilisent des langages et des formalismes différents pour exprimer leurs processus.

Cette situation combine donc trois difficultés essentielles : expliciter la sémantique des sources de données, expliciter la sémantique des processus métiers, et construire un entrepôt de données adapté. Nous détaillons à présent ces difficultés, qu'il faudra lever pour la réussite de ce projet.

## 1.3 Difficultés et feuille de route

Les solutions que nous avons envisagées couvrent plusieurs aspects :

(i) La construction d'une ontologie de domaine afin d'expliciter la sémantique des sources

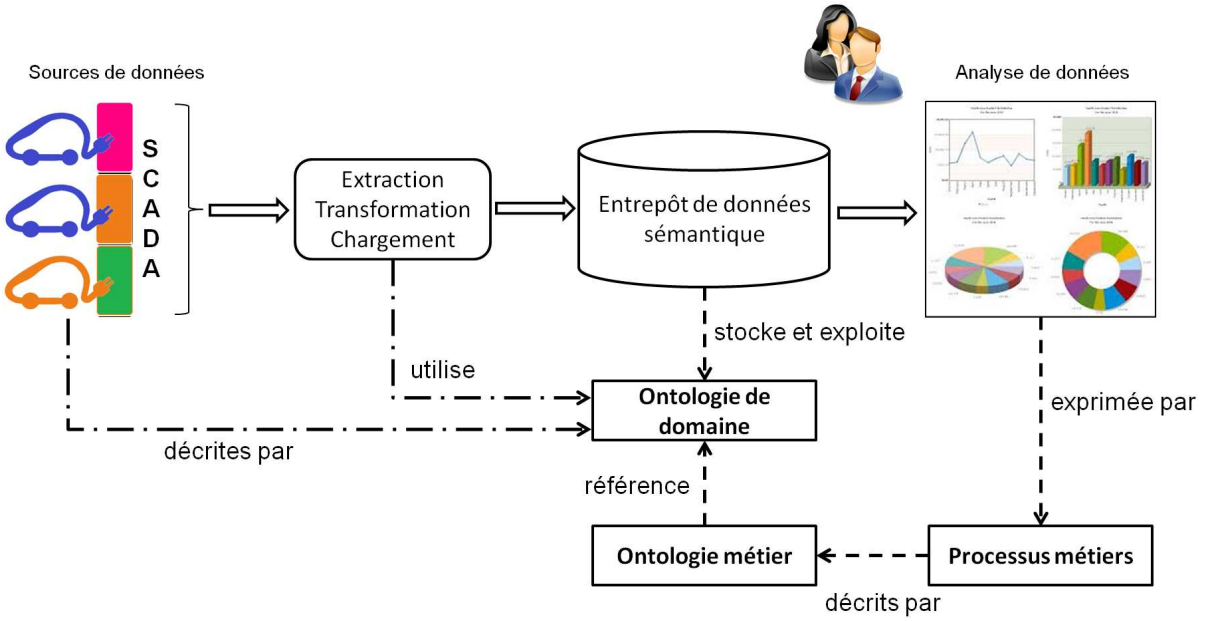


FIGURE 2 – Notre démarche

de données et permettre aux ingénieurs et aux experts d'EDF de manipuler l'entrepôt de données final en faisant abstraction de son implémentation physique. La construction d'une ontologie ex nihilo est une tâche difficile et coûteuse. Malgré tout, cette construction passe par un cycle de vie bien identifié comprenant les phases suivantes : (a) recueil des besoins, (b) spécification, (c) conceptualisation, (d) implémentation, (e) tests et (f) exploitation. Pour aborder cette situation, nous avons proposé une construction par brique de l'ontologie, qui sera exploitée par le cycle de vie de l'entrepôt de données, afin que nos solutions soient validées par EDF.

(ii) Une fois construite, l'ontologie est alors exploitée par l'entrepôt de données. Plusieurs travaux et projets de construction de systèmes d'intégration ou d'entrepôts de données [7] à partir d'ontologies existent. Nous pouvons citer le projet *PICSEL* [43] développé au sein du Laboratoire d'Informatique de l'Université de Paris X, *OBSERVER* [83], *COIN* [45], *OntoDawa* [147], *OntoDW* [63], ces derniers ayant été développés au sein du laboratoire (LIAS [8]). La construction d'un entrepôt de données est également une tâche coûteuse dont le cycle de vie respecte un ensemble de phases bien identifiées : (a) l'analyse des besoins, (b) la phase conceptuelle, (c) la phase logique, (d) la phase *ETL* (Extract, Transform, Load), (e) la phase de déploiement et (f) la phase physique. La construction incrémentale de l'ontologie pourra impacter la construction de l'entrepôt de données car, à chaque évolution de l'ontologie, l'entrepôt doit être régénéré. Pour éviter ce scénario, nous avons proposé la génération d'une nouvelle version de l'entrepôt à chaque brique ajoutée.

(iii) Pour formaliser les processus métiers, nous proposons l'utilisation de Business Process

7. Un entrepôt de données est un cas particulier d'un système d'intégration.
8. Laboratoire d'Informatique et d'Automatique pour les Systèmes

Model and Notation, où l'ensemble des concepts utilisés sont explicités via une ontologie dite de métier. Cette dernière est construite en respectant la même démarche que pour l'ontologie de domaine.

Au final, notre entrepôt stocke quatre composantes : les données, l'ontologie des données, les processus et l'ontologie des processus.

# 2 Contributions

Ce travail a donné lieu à trois contributions essentielles décrites dans la figure 3. Ces trois contributions ont été appliquées à la problématique de facturation d'EDF.

1. La première contribution, motivée par les attentes de l'entreprise EDF, concerne la proposition d'une méthode de création d'ontologie modulaire et incrémentale. Cette approche permet principalement la conceptualisation d'un domaine en évolution. En effet il devient alors possible d'étoffer progressivement l'ontologie avec de nouveaux modules, ou d'en remplacer certains par des modules plus pertinents. A partir d'un noyau de modules, les acteurs du domaine peuvent commencer à exploiter l'ontologie sans attendre une éventuelle version finale.
2. La présence de cette ontologie permet alors de concevoir un entrepôt de données sémantique. L'ontologie joue le rôle d'un schéma global partagé par l'ensemble de sources, d'où la réduction de l'hétérogénéité qui pourrait exister entre les sources participant au processus d'intégration. L'ontologie est également exploitée par l'ensemble des phases du cycle de vie de l'entrepôt de données. Elle jouera le rôle du modèle conceptuel sur lequel les décideurs peuvent interroger l'entrepôt. Les différents algorithmes d'$ETL$ seront alors définis au niveau de l'ontologie, contrairement aux approches traditionnelles, en encapsulant tous les aspects d'implémentation des bases de données sources. Notons également que l'$ETL$ prend en compte l'évolution de l'ontologie, en effet les versions de cette dernière sont des paramètres des algorithmes dédiés à l'$ETL$. Notre entrepôt de données est alors déployé sur une architecture de bases de données sémantiques qui comprend à la fois les données et leur ontologie. Cette architecture est dotée d'un langage de requêtes, appelé *OntoQL*, développé au sein du LIAS, qui permet l'interrogation des données directement via l'ontologie.
3. La dernière contribution porte sur la formalisation et l'explicitation de la sémantique des processus métiers. Une ontologie associée à ces processus a été construite et intégrée à l'entrepôt de données. Ainsi, notre entrepôt manipulera à la fois les données des véhicules électriques et les processus métiers nécessaires aux experts.
4. Dans le cadre de l'exploitation de l'entrepôt de données généré, nous avons proposé à EDF un système de tarification dynamique en utilisant la théorie des jeux, où deux joueurs sont identifiés : le client et l'entreprise EDF.

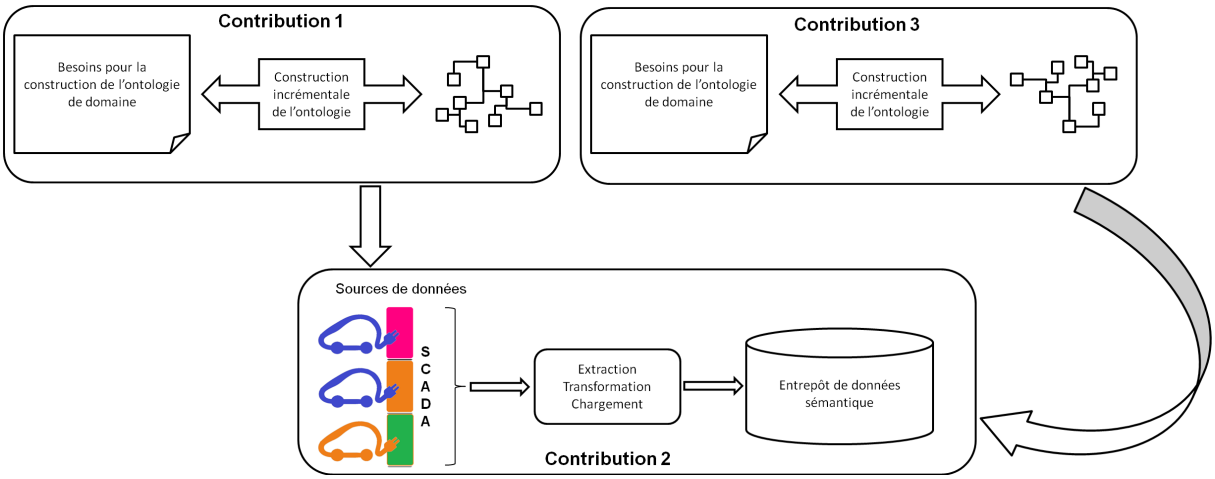


FIGURE 3 – Contributions des travaux

# 3 Organisation du mémoire

Ce mémoire est organisé en trois parties.

La première partie, présentant les états de l'art, comprend deux chapitres.

Le premier chapitre présente dans un premier temps un état des lieux sur les ontologies, leur modèles, leur classification. Une comparaison entre les modèles conceptuels et les ontologies est également donnée. Dans un deuxième temps, les différentes approches de construction d'ontologie sont décrites ainsi que les ontologies modulaires. Les cycles de vie de construction d'ontologie sont décrits, en mettant en évidence l'ensemble de leurs phases.

Le deuxième chapitre est lui consacré à la technologie d'entreposage des données. D'abord, des définitions et des concepts fondamentaux autour des entrepôts de données sont proposés. Le cycle de vie de conception d'entrepôt de données avec ses phases est présenté. Une classification sur le type des entrepôts de données comprenant les entrepôts de données traditionnels et les entrepôts de données sémantiques et leurs implémentations est également présentée. Finalement, nous concluons ce chapitre par l'étude de l'impact de la connexion des ontologies aux entrepôts de données.

La deuxième partie de ce manuscrit présente l'ensemble de nos contributions. Elle contient les trois chapitres suivants :

Le chapitre 3 est consacré au processus de construction de notre ontologie de domaine pour la partie sources de données. Ce processus est basé sur la notion de briques ontologiques, où chaque brique concerne un seul concept identifié par l'entreprise EDF. Un mécanisme d'assemblage des briques est décrit en utilisant des règles à base de liens entre les briques. Cette construction est illustrée avec le cas de la mobilité électrique.

Le chapitre 4 explicite notre construction de l'entrepôt de données sémantique avec une

description détaillée de l'ensemble des phases de son cycle de vie. Cette construction a été réalisée en utilisant la plateforme *OntoDB* développée au LIAS. Elle permet le stockage à la fois des données et de leur ontologie. L'ensemble des algorithmes d'$ETL$ sont réécrits en utilisant le langage *OntoQL* accompagnant cette plateforme.

Dans le chapitre 5, nous nous focalisons sur la capitalisation des connaissances et les processus métiers des utilisateurs d'EDF. Cette capitalisation se fait grâce à la connaissance des processus métiers que possède les ingénieurs et les experts d'EDF. Une modélisation des processus métiers à l'aide de Business Process Modelisation and Notation est alors donnée. Une implémentation de l'ensemble des processus ainsi que leur ontologie dans la plateforme *OntoDB* est décrite.

La troisième partie présente, en deux chapitres, une mise en œuvre de notre solution pour EDF et donne une conclusion générale de ce travail. Nous y esquissons également diverses perspectives.

Ces travaux ont donné lieu à trois publications :

- K. Royer, L. Bellatreche, S. Jean, One Semantic Data Warehouse Fits both Electrical Vehicle Data and their Business Processes, Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC 2014), Qingdao, China, October 8-11, 2014 ([112]).
- K. Royer, L. Bellatreche, S. Jean, Combining domain and business ontologies in a modular construction method: EDF study case, in Proceedings of the 38th International Convention, Business Intelligence Systems (MIPRO), pp. 1452-1457, IEEE, Croatia, May 26-30, 2014 ([111]).
- K. Royer, L. Bellatreche, A. Le Mouel, G. Schmitt, Un Entrepôt de Données pour la Gestion des Véhicules Électriques : Retour d'Expérience, 8e Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA'2012), pp. 118-127, Hermann RNTI, Bordeaux, 12-13 juin 2012 ([113]).

# Première partie

# États de l’art

Chapitre 1

# État de l'art sur les ontologies

## Sommaire

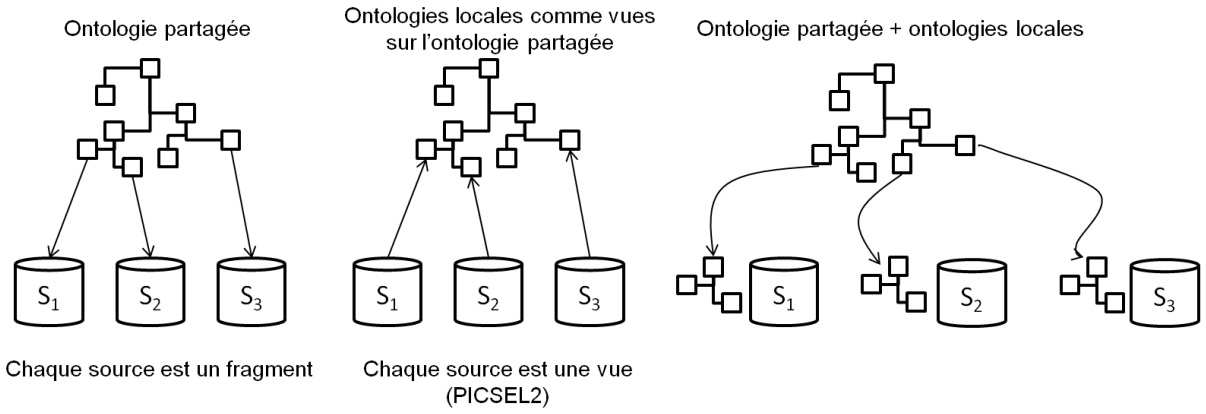


FIGURE 1.1 – Différentes architectures de systèmes d'intégration ontologiques

# 1 Introduction

L'ontologie est une branche de la philosophie qui étudie ce qui *est* afin de définir les propriétés générales de tout ce qui existe. Les ontologies ont été largement utilisées par un nombre important de communautés comme les bases de données, l'intelligence artificielle, la médecine, le traitement de langue naturelle, etc. Une ontologie est définie comme la représentation formelle et consensuelle, au sein d'une communauté d'utilisateurs, des concepts propres à un domaine et des relations entre ces concepts.

Les ontologies ont largement contribué à la construction des systèmes d'intégration matérialisés ou médiateurs pour réduire l'hétérogénéité syntaxique et sémantique d'une manière automatique. La sémantique du domaine est ainsi spécifiée formellement à travers des concepts, leurs propriétés ainsi que les relations entre les concepts. La référence à une ontologie par ces systèmes permet d'éliminer automatiquement les conflits sémantiques entre les sources. La connexion entre les sources participant au processus d'intégration et l'ontologie peut se faire de plusieurs manières : définition des termes de la source de données par les concepts ontologiques (projet *BUSTER* [127]), annotation des données (projet *SHOE* [78]), enrichissement de la source par des règles logiques (projet *PICSEL* [43]), etc. Trois principales architectures de systèmes d'intégration de données à base ontologique existent, comme le montre la figure 1.1, [143]:

- L'architecture avec une ontologie partagée où il n'y a qu'une seule et unique ontologie.
- L'architecture avec des ontologies locales où chaque source possède sa propre ontologie développée indépendamment des autres sources. Des correspondances entre les ontologies locales et partagée doivent être établies.
- L'architecture hybride où chaque source possède sa propre ontologie. Toutefois, toutes les ontologies locales sont mises en relation avec une ontologie partagée.

Dans ce chapitre, nous commençons par présenter les concepts fondamentaux liés aux ontologies ainsi que le cycle de vie de leur conception.

# 2 Définition

## 2.1 Origine des ontologies

C'est au XVII$^{e}$ siècle que le terme *ontologie* est apparu. En philosophie aristotélicienne il sert à désigner la branche de la philosophie qui étudie les propriétés les plus générales de l'être, telles que l'existence, la durée ou le devenir. C'est donc, selon le dictionnaire Larousse, la théorie de l'être. Le Dictionnaire de la Philosophie (Encyclopædia Universalis) indique que l'ontologie est *«l'étude des êtres en tant qu'être»* par opposition avec l'étude des êtres tels qu'ils nous apparaissent. Cette discipline s'attache donc à caractériser la nature profonde des êtres par opposition à l'étude de leur apparence ou de leurs attributs séparés.

Ce terme est également employé en médecine, cela désigne alors une doctrine qui vise à étudier l'être de la maladie comme s'il existait indépendamment du reste. Là encore, on retrouve la notion de définition de ce qui est, de l'essence du sujet étudié.

Avant même de détailler cette notion, notamment du point de vue informatique, on retrouve dans ses origines la clé permettant une exploitation industrielle d'une ontologie. En définissant un domaine en dehors du cadre applicatif mais uniquement pour en décrire l'essence et les composants on obtient un produit servant de référence. Cette idée est à la base de la notion d'ontologie en informatique.

## 2.2 La notion d'ontologie en informatique

Plusieurs définitions ont été proposées au fur et à mesure du développement et de l'enrichissement des ontologies. Nous avons retenu la définition suivante, proposée par *G. Pierra* [97] : une ontologie est «*une représentation formelle, explicite, référençable et consensuelle de l'ensemble des concepts partagés d'un domaine sous forme de classes, de propriétés et de relations qui les lient*». Les termes clés de cette définition sont les suivants :

- **Formelle** : grâce au formalisme employé pour décrire les concepts et leurs propriétés une machine est capable de traiter le langage dans lequel est défini l'ontologie. Les différents langages disponibles pour créer une ontologie sont présentés dans la suite de ce chapitre ainsi que leurs spécificités.
- **Explicite** : l'ensemble des éléments de l'ontologie (les concepts et les propriétés) sont spécifiés explicitement, c'est-à-dire indépendamment du contexte, ou d'éléments implicites. Aucun pré-requis n'est donc nécessaire pour comprendre le contenu de l'ontologie. Cet aspect impose un travail supplémentaire aux concepteurs mais en retour permet une réutilisation et un meilleur partage.
- **Référençable** : chaque concept de l'ontologie peut être référencé de manière unique afin d'expliciter la sémantique de l'élément référencé.
- **Consensuelle** : le plus important, cela signifie que l'ontologie est partagée dans son inté-

gralité par toutes les parties prenantes travaillant avec. Cela permet l'échange d'informations sans ambiguïté et la mise au point, par exemple, de services compatibles entre les parties prenantes.

Cette définition est une extension de celle donnée dans [51] : «*une ontologie est la spécification d'une conceptualisation*». Disposer d'une ontologie permet donc d'utiliser un langage commun sur un domaine qui n'est pas spécifique à un acteur du domaine.

Ainsi tous les modèles d'ontologies reposent sur les mêmes notions, d'abord les **concepts** du domaine qui sont modélisés par des **classes** et leurs **attributs**, puis les **relations** entre les concepts et enfin les **axiomes**. Voici quelques précisions sur ces éléments :

- **Concept** : un concept est une entité composée de trois éléments distincts.
  - Le **terme** : c'est l'expression du concept en langage naturel.
  - La **notion** : c'est la signification du concept.
  - Les objets dénotés par le concept : ce sont les **réalisations** ou **extensions** du concept.

  Les concepts peuvent être répartis en deux catégories [50].
  - Les **concepts primitifs** : ce sont les concepts de base de l'ontologie à partir desquels d'autres concepts peuvent être définis. L'ontologie ne fournit pas nécessairement de définition complète du concept car l'usage de ce concept est partagé par les parties prenantes.
  - Les **concepts définis** : ils disposent d'une définition complète dans l'ontologie. Ils peuvent également être définis par d'autres concepts (primitifs ou définis).
- Les **classes** : au cœur de l'ontologie, elles décrivent les concepts d'un domaine. Une classe peut avoir des **sous-classes** qui représentent des concepts plus précis. De la même façon, elle peut avoir une **super-classe** dont elle précise certains aspects. On désigne par une **base de connaissances** une ontologie et l'ensemble des instances des classes.
- Les **attributs** : ils décrivent les propriétés des classes.
- Les **relations** : elles désignent les associations entre les concepts de l'ontologie.
- Les **axiomes** : ce sont de assertions acceptées comme vraies dans le domaine étudié. Les axiomes et les règles permettent deux opérations : la vérification de la cohérence de l'ontologie et l'inférence de nouvelles connaissances.

Ces éléments sont communs à toutes les ontologies. A partir de cette base les ontologies ont été spécialisées pour s'adapter à différents domaines à analyser ou bien supporter des utilisations particulières.

## 2.3 Classifications des ontologies

Il existe plusieurs classifications des ontologies suivant leurs utilisations [134]. On peut citer une de ces classifications, proposée par [105, 52], où les ontologies sont classées en fonction de leur niveau de conceptualisation :

- Les **ontologies globales** (*top-level ontologies*) : ce sont des ontologies formelles, qui s'attachent à représenter un haut niveau de généralité et d'abstraction par rapport au do-

maine concerné, comme le projet KRAFT [140]. Grâce à un développement systématique et consensuel, elles permettent le partage de connaissances et leur transfert d'un contexte à un autre. Elles servent également de base aux développements d'ontologies plus concrètes.

- Les **ontologies de domaine** : ces ontologies s'attachent à décrire un domaine précis (géographie [38], médecine [124], énergie [100], etc.). Le niveau d'abstraction est moins élevé que dans les ontologies globales, elles vont apporter une spécialisation des concepts généraux des ontologies globales.
- Les **ontologies d'application** : ce sont des ontologies spécifiques à un champ d'application dans un domaine donné, ce dernier pouvant être décrit par une ontologie globale ou de domaine. On peut citer [33] où une ontologie de la cuisine a été mise au point pour permettre une utilisation harmonieuse de plusieurs services : application QooQ pour tablettes, le portail Cuisine AZ, les cours proposés par le site Cuisinix, etc. Un autre exemple est [28] pour des services de covoiturage.

Nos travaux s'intéressent aux ontologies de domaine, la section suivante présente une classification précise de ces ontologies.

Pour EDF l'objectif de l'ontologie est évidemment de répondre à des contraintes opérationnelles (intégration de données, échanges de données, requêtes, etc.). Cependant il existe aussi un niveau plus abstrait relatif aux sous domaines de la mobilité électrique ($ME$) qui correspondent en partie au découpage des différents services du groupe EDF. On aura, par exemple, la branche commerciale, la production d'électricité et le service Recherche et Développement (R&D). L'objectif d'EDF à ce sujet est partagé par les acteurs de la $ME$ mais sur leurs domaines respectifs. Les constructeurs de bornes de recharge, par exemple, sont peu intéressés par des considérations sociologiques et réciproquement les pouvoirs publiques, cherchant à réaliser des études sociologiques, ne s'intéressent pas aux détails des équipements de recharge. La dualité abstraite-domaine est une considération partagée : chacun veut que son champ soit bien décrit et ne disposer que de notions abstraites pour les autres domaines.

Par rapport à cette classification classique (ontologies globales, de domaine ou d'application), nous nous situons sur les deux premières classes : notre ontologie doit être à la fois globale pour tout couvrir et suffisamment détaillée pour que chaque acteur y trouve les informations qui l'intéressent.

## 2.4 Taxonomie des ontologies de domaine

Il convient de citer une seconde classification, ou plutôt une taxonomie, des ontologies qui s'organise autour de la description d'un domaine. Les ontologies sont principalement exploitées dans les domaines de la linguistique, des bases de données et de l'intelligence artificielle. Dans ces domaines l'usage de ces types d'ontologies diffère par la manière de conceptualiser le domaine étudié. Certaines ontologies vont chercher à décrire les mots et les relations entre

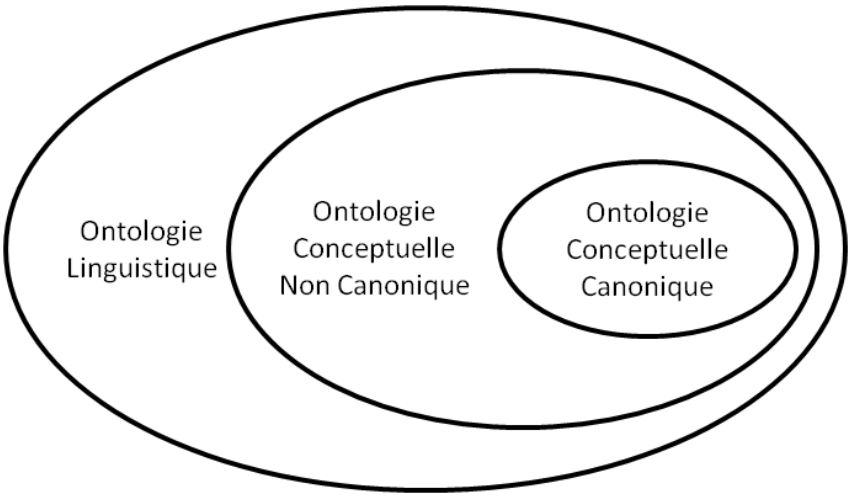


FIGURE 1.2 – Inclusion des ontologies

eux, comme la synonymie, l'antonymie, etc. Ce sont des ontologies dites *ontologies linguistiques*. D'autres vont chercher à avoir une, et une seule, manière de décrire un concept, ce sont les *ontologies canoniques*. Pour les bases de données, on parle *d'ontologie conceptuelle* pour manipuler des concepts et non des mots.

On distingue alors :

- Les *ontologies conceptuelles canoniques* (*OCC*) : les concepts décrits et utilisés sont des concepts primitifs. Ils possèdent une représentation unique et indépendante des autres concepts. Ils sont utiles pour la conception de base de données afin d'éviter les redondances et de permettre la création de formats d'échanges.
- Les *ontologies conceptuelles non canoniques* (*OCNC*) : elles utilisent des concepts primitifs et des concepts définis. Les concepts peuvent donc être reliés par des relations d'appartenance, d'équivalence, etc., ce qui permet de réaliser des déductions. Ces ontologies sont à ce titre utilisées dans les domaines de l'intelligence artificielle.
- Les *ontologies linguistiques* (*OL*) sont des ontologies qui visent à représenter les mots utilisés dans un domaine particulier. Les ontologies linguistiques permettent de fournir une représentation en langage naturel des concepts d'un domaine.

Dans cette classification, il existe une inclusion des classes illustrée par la figure 1.2. A partir d'une ontologie canonique on peut créer une ontologie non canonique en combinant des concepts primitifs. Par exemple : dans la *ME* on dispose de concepts de base comme *véhicule* et *utilisateur*. En les combinant on peut obtenir une *flotte* qui est composée de ces concepts de base. En créant ce nouveau concept on ne dispose plus d'une ontologie canonique. Pour poursuivre l'exemple, pour disposer d'une ontologie linguistique on pourra ajouter des éléments comme la désignation de concepts dans une autre langue, par exemple *fleet*, *flota*, etc.

Le travail de définition d'un domaine n'est pas étranger à une entreprise comme EDF. En effet de nombreux experts ont été amenés à travailler dans des groupes de normalisation (ces

normes sont citées dans le chapitre 3). Cette expérience a permis d'approcher les questions de taxonomie. Bien que le travail de normalisation soit différent de celui de la création d'une ontologie, il faut tout de même choisir les éléments qui seront dans la norme (des éléments primitifs uniquement ou également des éléments composés), le niveau de langage à utiliser (purement technique ou vulgarisé), les langues employées, etc. Du fait de cette expérience, nos travaux se sont tout de suite orientés vers la création d'une ontologie conceptuelle non canonique et ce pour plusieurs motifs. (1) La $ME$ est un domaine complexe, où de nombreux éléments sont la somme de plus petits composants qui n'existent que pour être assemblés. Par exemple : un utilisateur (concept fondamental) est composé d'un $\mathcal{VE}$ qui possède une batterie particulière, possède un moyen de s'identifier, effectue des charges et réalise des trajets. Il est donc nécessaire de disposer d'éléments primitifs, comme les moyens d'identification, mais également de l'assemblage d'éléments qui compose l'utilisateur. (2) De plus ce type d'ontologie supporte des mécanismes de raisonnements, or, vue l'inégalité de la qualité des sources, EDF et ses partenaires, envisagent de pouvoir déduire les données manquantes. Il est donc nécessaire d'utiliser une ontologie permettant ce genre de mécanismes.

# 3 Représentations des ontologies

Pour représenter une ontologie, d'un point de vue informatique, il faut recourir à un formalisme, c'est-à-dire un modèle permettant de représenter un ontologie. Souvent présenté sous la forme de modèle objet, le formalisme se compose d'entités et d'attributs permettant de décrire les constructeurs d'une ontologie pour les classes et les propriétés. De nombreux formalismes sont apparus comme $PLIB$ [96], $OWL$ [122], $DAML-OIL$ [29], $RDFS$ [142], etc. Ces modèles se différencient par l'objectif qu'ils visent, certains se focalisent sur la précision et l'unicité des éléments décrits donc les $OCC$, comme $PLIB$ ou $RDFS$. D'autres se concentrent sur les correspondances entre les vocabulaires pour permettre des déductions ou des inférences, c'est le cas du formalisme $OWL$ par exemple, pour les $OCNC$. Dans cette partie nous présentons les formalismes qui nous intéressent dans nos travaux sur les ontologies de domaine : $RDF(S)$ utilisé pour les ressources web, $DAML-OIL$ comme prédécesseur d'$OWL$ et $PLIB$.

## 3.1 Le formalisme RDFS

Le *Ressource Description Framework (RDF)* [142] : l'information sur le Web étant compréhensible seulement au niveau lexical-syntaxique, les outils logiciels indépendants ne sont pas en mesure de traiter l'information en l'absence des méta-informations. Ceci a amené le $W3C$ (World Wide Web Consortium) à élaborer une couche supplémentaire au dessus de $XML$ (eXtensible Markup Language) appelée Ressource Description Framework ($RDF$) pour traiter ces méta-informations. Le $W3C$ a développé $RDF$ un langage d'encodage de la connaissance sur les pages Web pour rendre cette connaissance compréhensible par les agents électroniques qui

effectuent des recherches d'informations. *RDF* permet de décrire des ressources simplement et sans ambiguïté, notamment les ressources web utilisées par de nombreux sites marchands. Toute ressource est décrite par des phrases minimales composées d'un sujet, d'un verbe et d'un complément, on parle alors de déclaration *RDF*.

*RDF Schema ou RDF(S)* [20] : le langage *RDF* ne propose pas de constructeur pour la conception d'ontologies. Il a donc été étendu par de nouveaux constructeurs pour permettre la définition d'ontologies, ce qui a donné lieu au modèle *RDF(S)*. C'est devenu une base du web sémantique car il permet de construire des concepts définis à partir de concepts présents à travers le web.

## 3.2 Le formalisme DAML+OIL

*DAML + OIL (DARPA*[9] *Agent Markup Language + Ontology Inference Layer)* [125, 29] : *DAML + OIL* a été créé suite à la fusion de deux travaux d'équipes de recherche qui sont : DARPA Agent Markup Language (*DAML*) du ministère de la défense américain, et Ontology Inference Layer (*OIL*) développé par la communauté de recherche européenne [37]. *DAML* est un langage de représentation qui fournit une sémantique formelle pour l'information. *OIL* a enrichi *RDF(S)* en offrant de nouvelles primitives permettant de définir les classes comme l'union de classes, l'intersection de classes et le complémentaire d'une classe. La fusion des deux langages a donné lieu à *DAML + OIL*, écrit en *RDF*, permettant d'enrichir le pouvoir d'expression de *RDF(S)*. Le *W3C* a utilisé le *DAML + OIL* comme base pour la construction de son propre langage d'ontologies : le langage *OWL*.

## 3.3 Le formalisme OWL

Le langage *Ontology Web Language* (*OWL*) [122, 94] est reconnu aujourd'hui par le *W3C* comme le standard pour représenter des ontologies pour le web sémantique. *OWL* est inspiré de *DAML* (projet américain) et d'*OIL* (projet européen) pour permettre une description riche des ontologies et leur partage. Pour cela *OWL* a étendu le modèle *RDFS* [55] pour que les ontologies puissent être manipulées par des ordinateurs et des humains, les opérateurs ajoutés permettent à l'ontologie d'être plus expressive et offre des capacités de raisonnement supérieures. Et *OWL* utilise une syntaxe *RDF/XML*.

*OWL* possède trois sous-modèles : *OWL – Lite*, *OWL – DL* et *OWL – Full*, chacun présente un compromis différent entre les capacités d'expression et la décidabilité. Le langage *OWL – Lite* permet de décrire une hiérarchie de classifications avec des contraintes simples, comme la cardinalité qui peut valoir 0 ou 1 uniquement. Pour avoir une capacité d'expression on peut utiliser *OWL – DL* qui permet de conserver le calcul des inférences, *OWL – DL* est fondé sur la logique descriptive et il intègre toutes les structures de langage *OWL*. Quant à *OWL – Full* il

9. Defense Advanced Research Projects Agency

permet aux utilisateurs de disposer d'une expressivité encore plus importante ainsi que la liberté syntaxique de *RDF* mais sans les capacités d'inférences. Ces modèles respectent la hiérarchie suivante : une ontologie $OWL-Lite$ valide est une ontologie $OWL-DL$ valide et une ontologie $OWL-DL$ valide est une ontologie $OWL-Full$ valide.

*OWL* est soutenu par le logiciel d'édition *Protege* [30] de l'université de Stanford.

## 3.4 Le formalisme PLIB

Le modèle Parts LIBrary (*PLIB*) [96] a été conçu initialement pour décrire différentes catégories de composants industriels et leurs instances. Son but est de permettre leurs échanges et leur modélisation de la façon la plus précise possible, par exemple on doit pouvoir retrouver à quelle classe appartient un objet, quelle propriété peut être appliquée à quels objets, etc. Pour que cela soit possible une ontologie *PLIB* doit être la plus précise possible. Dans *PLIB* les propriétés jouent un rôle essentiel car elles permettent d'associer à chaque concept un nom, une définition, etc. De fait les classes n'existent que comme domaine de propriétés (domaine de départ et domaine d'arrivée). *PLIB* permet également de créer des ontologies multi-lingues grâce à l'utilisation d'identifiants uniques qui vont servir dans les descriptions dans les différentes langues. Dans la mesure où une classe n'est créée que quand elle ne peut pas être définie par d'autres classes, c'est le modèle privilégié pour créer des ontologies conceptuelles canoniques.

Nous avons vu précédemment que notre travail s'orientait vers une ontologie conceptuelle non canonique. Comme notre domaine est amené à évoluer le langage choisi doit nous permettre de décrire précisément les relations, c'est pourquoi nous nous sommes orientés vers le langage *PLIB*.

## 3.5 Les ontologies dans le monde industriel

Les ontologies ont été largement utilisées dans le monde industriel, en particulier dans le domaine de l'ingénierie. On peut citer par exemple, l'ontologie normalisée IEC 61360-4 sur le domaine des composants électroniques, l'ontologie ISO 13399 sur le domaine des outils coupants, l'ontologie ISO 13584-501 sur le domaine des matériels de mesure, etc. Ces ontologies ont été acceptées comme des ontologies consensuelles et partagées et ce par tous les participants impliqués dans le processus *B2B* (Business to Business), où chaque fournisseur décrit les classes de ses composants dans son catalogue en référençant le plus possible l'ontologie normalisée. Dans ce cas, l'ontologie joue le rôle d'un dictionnaire. Sa présence facilite l'intégration et l'échange des composants entre les différents fournisseurs. Le LIAS, laboratoire dans lequel s'effectue cette thèse, a participé au développement et à la normalisation d'un nombre important d'ontologies dans le domaine de l'ingénierie (avionique avec Airbus, géologique avec l'Institut Français de Pétrole, etc.).

# 4 Ontologies vs. modèles conceptuels

Nous souhaitons décrire un retour d'expérience que nous avons eu au cours de discussions avec les décideurs d'EDF lors de l'élaboration de notre ontologie. Dès l'élaboration de l'ontologie de domaine pour EDF, nous avons constaté que les acteurs de ce projet ne font pas la différence entre les ontologies et les modèles conceptuels.

Les ontologies présentent des similitudes avec les modèles conceptuels classiques sur le principe de la modélisation car tous deux définissent une conceptualisation de l'univers du discours au moyen d'un ensemble de classes auxquelles sont associées des propriétés [123]. Les ontologies et les modèles conceptuels divergent cependant sur un point essentiel : l'objectif de modélisation qui est à l'origine de la contribution d'une ontologie dans une démarche de modélisation conceptuelle. Une ontologie est ainsi construite selon une approche descriptive, par opposition à un modèle conceptuel qui est construit selon une approche prescriptive. Une approche prescriptive a les implications suivantes [35] :

- seules les données pertinentes pour l'application cible sont décrites ;
- les données doivent respecter les définitions et contraintes définies dans le modèle conceptuel ;
- aucun fait n'est inconnu : c'est l'hypothèse du monde fermé ;
- la conceptualisation est faite selon le point de vue des concepteurs et avec leurs conventions ;
- le modèle conceptuel est optimisé pour l'application cible.

Ces caractéristiques, dues au fait qu'un modèle conceptuel dépend fortement du contexte dans lequel il a été conçu, sont à l'origine des hétérogénéités identifiées lors de l'intégration et de l'échange de données issues de sources hétérogènes [76]. Contrairement à un modèle conceptuel qui prescrit une base de données selon des besoins applicatifs (orientés application), une ontologie est développée selon une approche descriptive (orientée domaine). Elle permet de décrire les concepts et les propriétés d'un domaine donné indépendamment de tout objectif applicatif et de tout contexte hormis le domaine sur lequel porte l'ontologie. En plus de la capacité de conceptualisation, les ontologies apportent d'autres dimensions comme : l'identification des concepts, la consensualité et le raisonnement.

- **Identification des concepts** : les concepts dans une ontologie possèdent des identificateurs universels (par exemple, un *URI* [10]) leur permettant d'être référencés par des applications externes. Ces identifiants universels permettent à chaque partenaire impliqué dans un projet avec EDF de savoir de quelle notion il s'agit indépendamment de sa désignation en langue naturelle. Ainsi chaque partenaire peut utiliser sa langue ou son vocabulaire préféré pour désigner un concept sans que cela pose de problème lors de l'intégration des données.
- **Consensualité** : l'aspect consensuel des ontologies facilite la tâche des concepteurs qui travaillent sur divers projets référençant les mêmes ontologies et qui souhaitent partager

10. *Universal Resource Identifier* : identifiant unique dont la syntaxe est normalisée

et échanger des données sur leurs modèles. Dans le contexte de nos travaux, les concepts partagés entre EDF et ses partenaires concernent les $\mathcal{VE}$ (batteries, données des charges, infrastructures de recharge, etc.).

– **Raisonnement** : les ontologies, de par leur aspect formel, offrent des mécanismes de raisonnement permettant de vérifier la consistance des informations et d'inférer de nouvelles données. Cette capacité permet par exemple de vérifier que chaque concept ajouté dans l'ontologie est consistant (c'est-à-dire qu'il peut avoir des instances) avec le reste de l'ontologie. Ceci est particulièrement utile dans notre contexte où nous souhaitons proposer une démarche de construction incrémentale d'une ontologie modulaire.

A la lumière de ces différences, il est par conséquent intéressant de considérer une ontologie comme un premier niveau de spécification d'un modèle conceptuel. Nous citons comme exemple les trois travaux suivants proposés par *Roldan-Garcia et al.* [40], *Sugumaran et al.* [129] et *Fankam et al.* [35] qui permettent de définir le modèle conceptuel d'une base de données à partir d'une ontologie supposée préexistante.

# 5 Constructions des ontologies

Dans cette section, nous détaillons l'ensemble des approches de construction d'ontologies ainsi que les ontologies modulaires.

## 5.1 Approches de construction d'ontologie

A partir de la définition d'une ontologie, indiquée au début du chapitre, il est possible de créer une ontologie à partir de rien [90] car la définition est suffisamment complète pour permettre une telle démarche. Avec le développement des ontologies dans diverses communautés et d'outils visant à les exploiter, des méthodes de création sont apparues.

Il existe deux grandes catégories de méthodes de création d'ontologie qui s'appliquent dans des cas différents. Une partie des méthodes disponibles se sert de documents et d'informations qui existent déjà, elles visent alors à extraire des informations de l'existant [108, 130]. Les autres méthodes encadrent la démarche de création à partir de rien, *i.e.* : avec des connaissances implicites du domaine.

Trois rapports, publiés récemment, recensent les différentes méthodes de construction d'ontologies [143, 12, 128]. Voici deux méthodes qui pourraient être appliquées à la $ME$.

### 5.1.1 Méthode globale

La première est aussi la plus ancienne [90, 60], cette *méthode globale* est réalisée en plusieurs étapes.

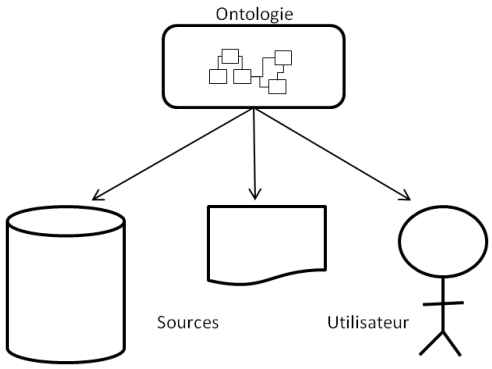


FIGURE 1.3 – Ontologie mise au point par la méthode globale

La première étape consiste à analyser le domaine étudié. Cette analyse est commune à toutes les méthodes, elle consiste à définir le **périmètre de l'ontologie**. En effet le but de l'ontologie étant d'être exhaustif sur un domaine, cette étape est cruciale pour dimensionner l'ontologie. Elle implique de faire des choix d'experts sur les éléments à inclure. Une fois le périmètre établi il faut définir la **granularité de l'ontologie**, c'est la deuxième étape du dimensionnement. Il s'agit là de décider de la précision des concepts utilisés, ce qui déterminera le moment à partir duquel un concept est considéré comme atomique. Cette fois il est nécessaire de faire intervenir les experts du domaine mais également les futurs utilisateurs pour s'assurer que les concepts seront exploitables par tous.

La seconde étape est la conceptualisation du domaine. Elle est directement liée à l'usage qui sera fait de l'ontologie et elle constitue la continuité des choix précédemment effectués sur la granularité des concepts. Prenons l'exemple de la $ME$ : pour EDF un $\mathcal{VE}$ est une batterie qui se déplace alors que pour un constructeur automobile un $\mathcal{VE}$ sera un concept complexe à décrire. La conceptualisation va également demander la création des liens entre les concepts.

La dernière étape va d'avantage concerner l'implémentation ou la représentation de l'ontologie. L'ontologie obtenue (voir figure 1.3) chapeaute l'ensemble des sources, systèmes d'information, utilisateurs, etc. Chacune des étapes citées requiert l'approbation de tous les utilisateurs de l'ontologie, c'est une démarche consensuelle globale.

La méthode globale a ouvert la voie à des travaux spécifiques sur la création des ontologies. On remarque qu'elle se base sur deux hypothèses :

1. Le domaine est bien défini.
2. Les utilisateurs peuvent aboutir à un consensus global.

Or le domaine que l'on cherche à décrire par une ontologie n'est pas clairement défini. Comme indiqué dans le chapitre précédent, le domaine de la $ME$ est récent et en pleine évolution. Créer une ontologie sur la connaissance du domaine d'aujourd'hui produira une ontologie qui deviendra obsolète en quelques mois. De plus, les contributeurs de l'ontologie sont très nombreux et internationaux. Il n'est donc pas possible de tous les réunir ou de les faire échanger afin

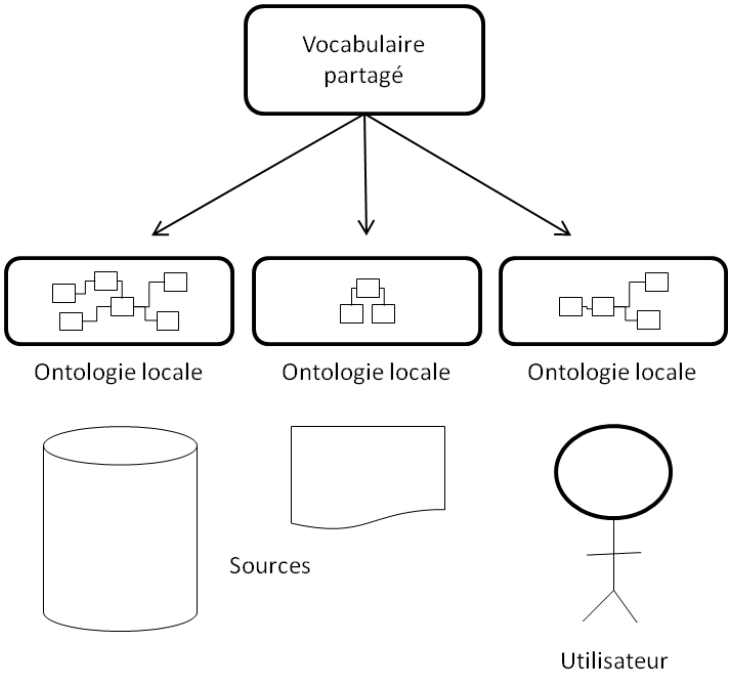


FIGURE 1.4 – Ontologies locales et vocabulaire partagé

d'atteindre un consensus global, d'où le développement de nouvelles méthodes moins contraignantes basées sur un vocabulaire commun.

### 5.1.2 Ontologies locales et vocabulaire commun

Cette seconde méthode est issue de la méthode globale décrite précédemment. Des ontologies locales, spécifiques à des sources ou des usages, sont créées (voir figure 1.4). Toutefois ces ontologies locales partagent un même vocabulaire. C'est-à-dire que si l'intersection des périmètres de deux ontologies locales est non-nul alors les concepts de l'intersection sont les mêmes dans les deux ontologies.

Cette méthode est plus souple que la précédente. En effet il est possible de définir localement une ontologie pour appréhender une source de données particulière ou pour satisfaire les besoins spécifiques d'un utilisateur.

Toutefois les domaines traités par les ontologies locales peuvent être hétérogènes et donc requérir des concepts spécifiques. On se retrouve alors avec une déclinaison d'un même concept pour décrire précisément le domaine. Ce qui oblige, ensuite, à créer des *mappings* entre les concepts nécessaires à chaque ontologie, c'est-à-dire définir les correspondances entre les concepts des ontologies. Formaliser les sources et les besoins est important mais l'objectif n'est pas encore atteint.

## 5.2 Ontologies modulaires

Les méthodes classiques de construction d'ontologies emploient des démarches globales : description directe de tous les concepts du domaine et consensus entre tous les acteurs. Cette démarche requiert plusieurs hypothèses fortes :

- Le domaine est connu dans son intégralité et il est stable, *i.e.* il n'est pas, ou très peu, sujet à des changements.
- Les acteurs du domaine amenés à utiliser l'ontologie peuvent se réunir et aboutir à un consensus. Ou alors, l'ontologie est développée par un nombre limité d'experts attentifs aux besoins des utilisateurs.

Ces hypothèses peuvent se retrouver au sein d'une entreprise, toutefois il existe de nombreux cas qui ne vérifient pas toutes ces hypothèses. A partir de ce constat des travaux visant à diviser les différentes démarches sont parus. Une des approches retenue consiste à travailler avec des assemblages d'ontologies locales : les ontologies modulaires.

### 5.2.1 Définition d'un module

Un module est un sous-ensemble de concepts qui peut être relié à d'autres modules. Un module possède un périmètre précis ce qui en fait une ontologie locale sur un sous-ensemble du domaine traité par l'ontologie.

Une approche proposée par [106] propose d'utiliser les modules ontologiques comme des API (*Application Programming Interface*). C'est-à-dire de créer une banque de concepts à partir desquels un utilisateur peut former son ontologie. Dans ce cas le module ne définit pas les liens avec les autres modules, c'est à l'utilisateur de les établir. Cette approche ressemble à l'une des méthodes présentées précédemment, ici le vocabulaire commun est remplacé par des concepts ontologiques.

### 5.2.2 Intérêts de la modularité

L'un des freins à la création d'une ontologie réside dans le besoin d'une validation globale des concepts. Avec la décomposition du domaine en sous-domaines, le processus de validation peut être accéléré dans la mesure où il y a moins d'intervenants par domaine.

Les moteurs d'inférences sont plus efficaces sur des ontologies réduites [33], ils peuvent donc s'exécuter plus facilement sur des modules, *i.e.* des ontologies locales au périmètre réduit, que sur toute l'ontologie.

Enfin la maintenance d'une ontologie locale est plus aisée (mise à jour ou remplacement). En effet il y a moins d'acteurs impliqués et moins de concepts à maintenir.

### 5.2.3 Stratégies d'assemblage

Il existe plusieurs types de collections de modules à partir des desquelles des ontologies vont être construites.

On peut considérer différentes ontologies comme une collection de modules sur différents domaines. Cette approche s'attache à réutiliser des ontologies existantes. Le projet [74] mis en place par la *Linked Data Community* reprend cette approche. L'avantage est d'exploiter des ontologies existantes maintenues indépendamment les unes des autres. Cette méthode ne cherche pas à ré-utiliser des ontologies, elle se focalise sur l'exploitation de ce qui existe pour former le plus grand corpus possible. Par contre la mise en place de cette méthode pour construire une ontologie globale pose les mêmes problèmes que précédemment : les acteurs doivent tous accepter d'utiliser les mêmes ontologies locales (supposées existantes), les ajouts ou modifications suivent le même processus de validation, etc. De plus il est difficile de trouver des ontologies répondant exactement aux besoins d'un acteur et *a fortiori* de tous les acteurs.

Une autre catégorie de collection est plus orientée vers la création. Les modules ont des périmètres restreints et leur existence est justifiée par leurs contributions à l'ontologie globale, contrairement à l'approche décrite précédemment. Il s'agit d'une démarche visant à créer une ontologie sur mesure. Ce type de collection peut également être définie avec une contrainte de disjonction [106]. Cette contrainte impose que l'intersection de deux modules quelconques est l'ensemble vide. Cela permet d'avoir des modules plus précis qui peuvent être maintenus indépendamment les uns des autres. De plus cette contrainte permet d'exploiter des moteurs d'inférences localement (sur chaque module).

### 5.2.4 Synthèse des approches

L'approche de construction classique où l'ontologie va être construite de A à Z à partir de la définition du domaine et des besoins des utilisateurs n'est clairement pas adaptée car aucune de ses hypothèses n'est vérifiée. En revanche les ontologies locales apportent des solutions : à partir d'un vocabulaire commun, chaque acteur monte sa propre ontologie et conserve une certaine capacité à échanger avec les autres. Dans la pratique ceci n'a pas aussi bien fonctionné. En amont des projets concernant le déploiement de $\mathcal{VE}$ (voir chapitre 6) les partenaires, dont EDF, se sont réunis. Ces réunions ont entre autre vu la création d'un format d'échange, depuis le format des fichiers de données $CSV$ [11] jusqu'à des formalismes $XML$ poussés. Les partenaires ont donc définis un vocabulaire commun et chacun disposait de son côté de ses modèles conceptuels et de ses applications. Au fil du temps le vocabulaire commun s'est perdu. Le domaine évoluant chacun a développé son nouveau vocabulaire et le cumul de petites altérations du vocabulaire initial a conduit à une grande hétérogénéité. Il manque donc à cette approche un réel intérêt à partager quelque chose entre les partenaires et, au delà, avec les autres acteurs du domaine.

11. *Comma-Separated Values* : format numérique où les données sont séparées par des virgules

D'autre part ces méthodes nécessitent toujours de réunir les acteurs pour avoir un terrain d'entente. Si cela s'avérait possible entre une poignée de partenaires pour un projet spécifique, il n'en est pas de même avec tous les acteurs du domaine et sur le long terme.

Les expériences industrielles d'EDF nous ont à nouveau fait avancer dans notre démarche de recherche. C'est ce qui nous a amenés à étudier les ontologies modulaires. Cette approche est intéressante car elle permet de ré-utiliser des éléments déjà existants : le vocabulaire commun précédemment cité ou les éléments des modèles conceptuels, bref elle permet à la fois de gagner du temps et de disposer de bases connues. L'idée de travailler avec des modules prend tout son sens lorsqu'il s'agit d'étudier un nouveau domaine comme celui de la *ME*. Le domaine est en cours de définition mais on peut envisager d'en définir les parties connues avec des modules et de préférence en s'appuyant sur des éléments existants.

Toutefois les stratégies d'assemblage existantes sont basées sur l'hypothèse que les acteurs peuvent se réunir et parvenir à un consensus. Comme indiqué plus tôt cela n'est pas envisageable sur la *ME*. Bien que le choix se soit porté sur une ontologie modulaire, il reste à créer une nouvelle méthode d'assemblage plus souple mais qui conserve la structure et la définition d'une ontologie. Pour travailler sur les modules nous nous sommes alors intéressés au cycle de vie des ontologies afin d'exploiter les méthodes existantes au niveau des modules.

# 6 Principaux cycles de vie de construction des ontologies

Les principales approches pour créer des ontologies s'appuient sur des cycles de vie de création. Cette section présente les étapes des cycles, les principaux cycles de vie que l'on retrouve ainsi que leurs hypothèses de mise en œuvre.

## 6.1 Étapes des cycles de vie

Les différents cycles de vie présentés dans les sections suivantes se basent sur un même ensemble d'étapes. Ils se distinguent ensuite par leurs façons d'enchainer ces étapes. Voici les étapes que l'on retrouve dans les cycles :

- *Initialisation* : la démarche de création d'une ontologie est initiée par des besoins communs à des acteurs d'un domaine, par exemple rendre compatible des produits ou des services.
- *Spécification* : spécifier un domaine correspond à s'interroger sur les besoins auxquels l'ontologie devra répondre. On va également définir le périmètre du domaine en restreignant les éléments à décrire. C'est aussi dans cette phase que les éléments existants vont être étudiés pour enrichir la réflexion et envisager de les ré-utiliser, et que le formalisme le plus à même de répondre aux besoins va être choisi.
- *Conceptualisation* : à partir des spécifications les concepts ontologiques seront concep-

tualisés. Cette phase va permettre de décrire de façon abstraite le concept contribuant à répondre aux besoins préalablement définis. La force de l'ontologie repose sur les concepts ontologiques et aussi sur le consensus des acteurs du domaine sur ces concepts. Le consensus doit se faire sur la conceptualisation des concepts.
- *Implémentation* : le concept va être implémenté avec le formalisme retenu lors de la phase de spécification.
- *Test* : lorsque cette phase est employée elle permet de tester l'implémentation de l'ontologie par rapport aux spécifications. En fonction des résultats des tests, les différents cycles de vie proposent différentes réponses. Ces réponses seront abordées dans la description de chaque cycle.
- *Exploitation* : c'est la phase finale du cycle de vie, la mise en œuvre de l'ontologie obtenue à travers les étapes du cycle concerné. L'exploitation d'une ontologie comprend aussi le travail de diffusion de l'ontologie pour amener de nouveaux acteurs du domaine à l'utiliser.

## 6.2 Différents cycles

### 6.2.1 Cycle en cascade

Ce cycle de vie est le plus classique [110], il s'agit de suivre une logique naturelle : on ne peut pas implémenter un élément qui n'est pas conceptualiser et on ne peut pas conceptualisé un élément qui n'est pas spécifié. Ce cycle se base sur plusieurs hypothèses.
- Le domaine est stable et connu, il peut donc être décrit de manière exhaustive et cette description ne changera pas.
- Les besoins sont connus et clairement définis.
- Les acteurs sont en mesure d'atteindre un consensus global, c'est-à-dire sur tous les concepts de l'ontologie.

La démarche est d'abord initialisée, ensuite l'ontologie puis ses concepts sont spécifiés. Une fois que tout est spécifié, chaque concept ontologique va être conceptualisé de façon à provoquer un consensus. Puis toute l'ontologie est implémentée à partir de la conceptualisation et livrée pour être exploitée par les acteurs du domaine.

La figure 1.5 résume ces étapes. Le modèle en cascade conduit donc des besoins vers un produit fini et opérationnel.

### 6.2.2 Cycle itératif

Le cycle itératif [133, 57] est une amélioration du cycle en cascade [70] pour délivrer un produit plus abouti, en outre il se base sur des hypothèses un peu plus faibles :
- Le domaine est stable et connu.
- Les besoins doivent être connus.

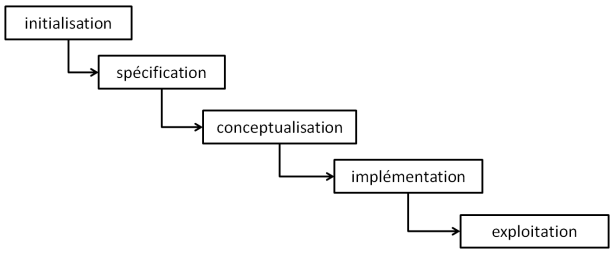


FIGURE 1.5 – Cycle de vie en cascade

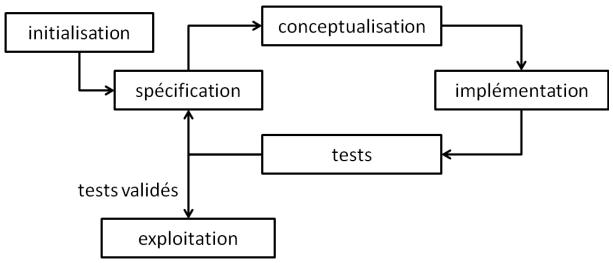


FIGURE 1.6 – Cycle de vie itératif

– Les acteurs sont en mesure d'atteindre un consensus global.

Une fois initié ce cycle passe par les mêmes étapes que le cycle en cascade : spécifications des besoins, conceptualisation des éléments de l'ontologie et leurs implémentations. Toutefois, une fois implémentée l'ontologie va être testée par les acteurs du domaine. Ces tests ont pour but d'affiner les besoins précédemment exprimés. Si les tests sont concluants alors l'ontologie créée par ce cycle peut être exploitée sinon le cycle recommence en révisant les spécifications.

Ce cycle permet de créer une ontologie mais également les concepts ontologiques s'ils peuvent être testés sans nécessiter toute l'ontologie (voir figure 1.6).

### 6.2.3 Cycle incrémental

Les cycles itératifs permettent d'obtenir un produit fini à partir d'hypothèses réduites comparées au cycle en cascade. Cependant l'ontologie ne pourra pas être exploitée tant que les itérations auront lieu. Ce constat peut être gênant lorsqu'il est nécessaire d'obtenir rapidement des éléments ontologiques exploitables. Le cycle incrémental propose une approche permettant de disposer d'éléments exploitables plus rapidement [73]. Il se base sur les hypothèses suivantes :

– Le domaine est stable et connu.
– Les besoins doivent être connus mais n'ont pas besoin d'être particulièrement précis ni définis.
– Les acteurs sont en mesure d'atteindre un consensus global.

Le principe est de se baser sur un noyau de concepts ontologiques. Ce noyau peut être développé à partir d'un cycle en cascade ou itératif (voir figure 1.7). Il est construit d'autant plus rapidement que le noyau est réduit.

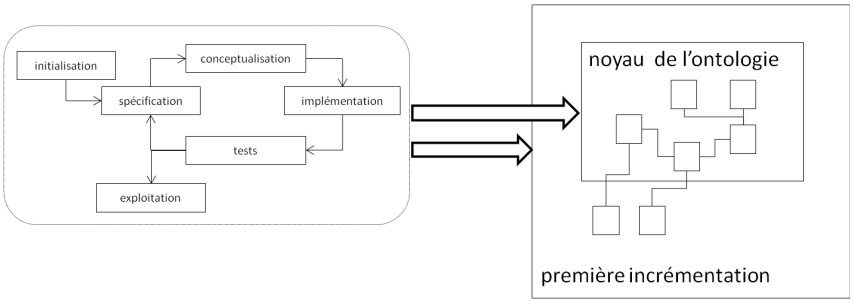


FIGURE 1.7 – Cycle de vie incrémental (avec incrémentation par un cycle en cascade)

Une fois le noyau développé il est mis en phase d'exploitation. En parallèle l'extension du noyau est entreprise, toujours selon l'un des cycles précédents.

### 6.2.4 Cycle en spirale et prototype évolutif

Dans le cas où les besoins sont amenés à changer au cours de la création de l'ontologie le cycle en spirale propose des mécanismes intéressants [16]. Il se base sur les hypothèses suivantes :

- Le domaine est stable et connu.
- Les besoins doivent être connus mais n'ont pas besoin d'être particulièrement précis ni définis et ils peuvent évoluer au cours de la création de l'ontologie.
- Les acteurs sont en mesure d'atteindre un consensus global.

Ce cycle combine un cycle en cascade avec une démarche d'évaluation. Une première version de l'ontologie est produite par un cycle en cascade, cette ontologie n'est pas nécessairement complète. Ce premier prototype est mis à l'épreuve par des tests afin d'établir ses forces, ses faiblesses et les coûts de la poursuite du cycle. Si les acteurs ne sont pas satisfaits, ou que le coût s'avère trop élevé pour qu'il soit intéressant de faire un cycle pour affiner l'ontologie alors la création est arrêtée. Sinon le cycle est recommencé en tenant compte de l'évaluation, le nouveau prototype sera évalué avec les mêmes critères pour en mesurer l'évolution, et ce jusqu'à ce que les acteurs soient satisfaits.

Ce cycle s'attache d'avantage à affiner un prototype en contrôlant les coûts des itérations (voir figure 1.8).

L'affinage d'un premier prototype sans évaluation précise conduit à l'approche de type prototype évolutif, plus rapide que le cycle en spirale mais potentiellement plus coûteuse.

## 6.3 Synthèse des principaux cycles et choix

Les cycles présentés ci-dessus exploitent les mêmes phases pour créer une ontologie avec des enchaînements différents mais surtout avec des hypothèses différentes. La figure 1.9 montre

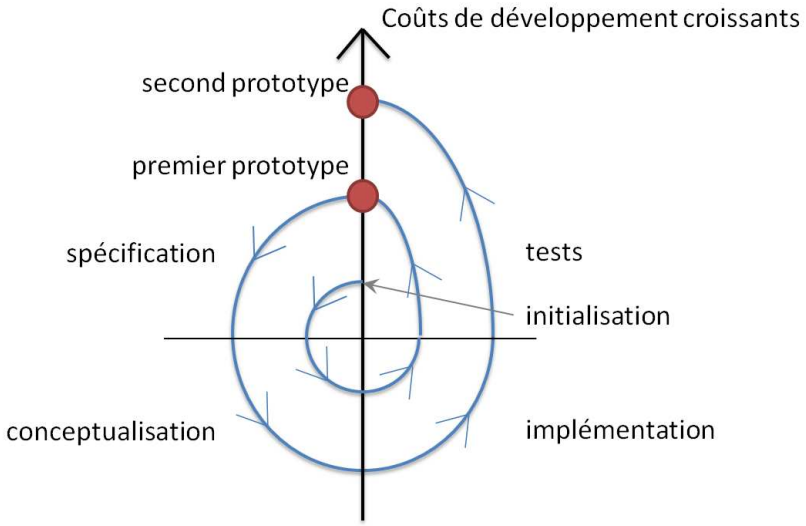


FIGURE 1.8 – Cycle de vie en spirale

les cycles à adopter en fonction de certains besoins et elle indique la force des hypothèses nécessaires à la mise en place de chaque cycle.

On observe qu'en fonction des besoins vis-à-vis de la création de l'ontologie plusieurs cycles sont disponibles suivant : la connaissance des besoins et le domaine, la disposition d'ontologies intermédiaires au cours de la création, le changement des besoins auxquels doit répondre l'ontologie, etc. Plus les contraintes sont fortes et plus le cycle de vie doit être souple et minimiser le coût des changements.

Comme indiqué, le domaine sur lequel nous avons travaillé avec EDF, la $ME$, est en cours de définition. Rien qu'au sein d'EDF plusieurs départements contribuent au développement de ce domaine, on peut citer les départements regroupant des experts en sociologie, en transport, en analyse de données ou encore sur des nouveaux modèles de batteries. Dans ce cadre on ne rentre dans aucun cycle car le domaine n'est pas connu dans sa totalité, il s'étend et se complexifie. D'autant plus que d'autres industriels effectuent la même démarche : les constructeurs de véhicules, les constructeurs de bornes ou encore les fournisseurs d'électricité des autres pays.

En dehors de la connaissance du domaine qui évolue, les hypothèses concernant les besoins sont vérifiées. Les besoins sont bien connus, même s'ils sont soumis à l'évolution du domaine, car ils correspondent à l'activité historique d'EDF mais appliquée à de nouveaux équipements. Par exemple le groupe EDF a besoin de disposer d'un minimum d'informations sur les charges des $\mathcal{VE}$ pour la facturation. Sur cet exemple les éléments du domaine sont connus : il faut connaître la référence du client, la borne utilisée, l'énergie chargée, la date de début de charge, la date de fin de charge, etc. Le besoin est clair, l'ontologie doit contenir ces informations et elles doivent être reliées entre elles. Et il est possible de concevoir une ontologie sur ces éléments sans savoir quelles nouvelles données seront disponibles à l'avenir.

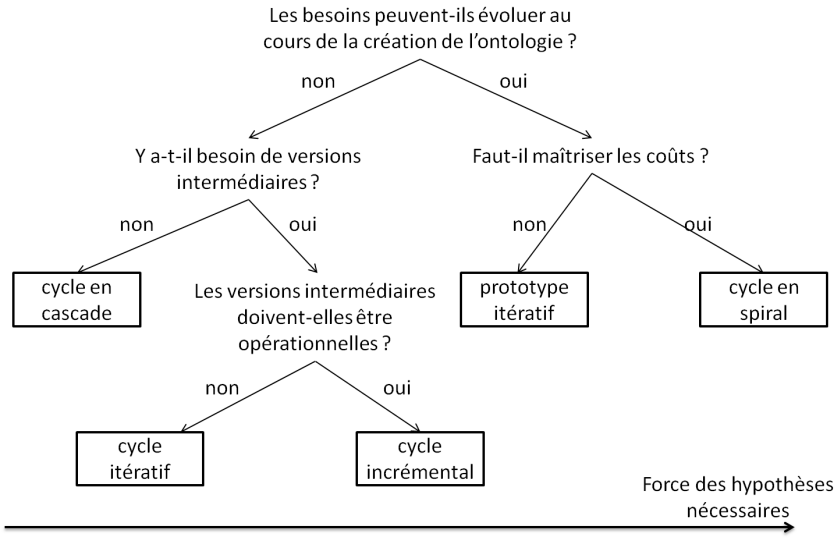


FIGURE 1.9 – Choix des cycles de vie et force des hypothèses requises pour chaque cycle

Dans cette situation il a fallu s'adapter. Nous avons vu dans la section précédente qu'une approche modulaire permettrait de créer l'ontologie sur notre domaine en pleine évolution. Or un module, ou un ensemble de modules, est développé sur une portion connue du domaine sur lequel des besoins précis sont exprimés. Ainsi la décomposition en module permet, comme dans l'exemple ci-dessus, de vérifier les hypothèses nécessaires à la mise en place des cycles.

En travaillant au niveau modulaire on s'affranchit de la nécessité de disposer d'éléments ontologiques (version de l'ontologie ou concepts ontologiques) intermédiaires. En reprenant l'arbre de la figure 1.9 : (1) les besoins n'évoluent pas lors de la création d'un module, (2) afin de bien définir le module (avec les différents experts) il est préférable d'avoir des versions intermédiaires et (3) les experts pouvant attendre la version finale d'un module, **le cycle recommandé est alors le cycle itératif**.

# 7 Bilan

Dans ce chapitre nous avons présenté les concepts fondamentaux d'une ontologie : sa définition, ses caractéristiques, etc. Plusieurs définitions sont proposées dans la littérature, dans notre cas nous retenons celle de *G. Pierra* [98] qui décrit une ontologie comme «*une représentation formelle, explicite, référençable et consensuelle de l'ensemble des concepts partagés d'un domaine en termes de classes et de propriétés*».

Suivant la nature des concepts à représenter dans une ontologie nous avons vu qu'il existe plusieurs classifications des ontologies. Les ontologies canoniques forment une base, comme en mathématiques, pour exprimer les concepts primitifs. Elles sont surtout utilisées en ingénierie pour référencer de manière unique un composant. Les ontologies non-canoniques ajoutent la

possibilité d'exprimer des concepts composés de plusieurs éléments de bases, cela permet donc d'exprimer des concepts plus complexes et, suivant les domaines, plus proche de la réalité. Enfin les ontologies linguistiques sont encore plus générales, elles peuvent apporter des synonymes, des antonymes, d'autres langues, etc.

Pour construire une ontologie appropriée au domaine, il existe deux grandes catégories de méthodes : l'exploitation de l'existant et la création à partir de rien. L'exploitation de documents existants permet de ré-utiliser une littérature sur un sujet grâce à diverses méthodes. La création d'une ontologie à partir de rien est un autre exercice, tout aussi difficile. Les méthodes s'attachent à réunir des experts pour établir le périmètre de l'ontologie, en décrire les concepts de façon exhaustive et arriver à un consensus général. Ces méthodes requièrent de pouvoir définir le domaine et d'en réunir les experts. Devant la difficulté à remplir ces hypothèses des travaux ont proposé des méthodes plus flexibles. Les ontologies modulaires font parties de la réponse, elles proposent plusieurs approches : le découpage du domaine en modules indépendants ou l'agrégation d'ontologies existantes sur des sous-domaines.

Nous avons également vu les différents formalismes pour décrire les ontologies. Des formalismes tels que *RDFS* ou *PLIB*, orientés gestion et échange de données, sont souvent utilisés dans l'ingénierie. *DAML+OIL* ou *OWL* se retrouvent dans le web sémantique et l'intelligence artificielle pour l'aspect inférence de données.

Pour résumer les choix effectués jusqu'à présent, au vu de la littérature et des précédentes expériences d'EDF, nous nous sommes orientés vers une ontologie à la fois globale et de domaine. L'ontologie sera une ontologie conceptuelle non canonique formalisée avec *PLIB*. Sa création pourra se faire de façon modulaire en développant les modules avec des cycles itératifs. Toutefois il nous faut proposer une méthode d'assemblage des modules pour respecter les contraintes du domaine : évolution du domaine avec une multitude d'acteurs et besoin opérationnel immédiat.

Le chapitre suivant propose un état de l'art sur les entrepôts de données qui sont au centre de notre solution globale. Nous nous attacherons à revoir les interactions entre les entrepôts et les ontologies afin de démontrer l'intérêt de mettre en place une nouvelle méthode de création d'ontologie avec EDF.

Chapitre 2

# État de l'art sur les entrepôts de données

## Sommaire

# 1 Introduction

Comme nous l'avons indiqué dans l'introduction générale, la technologie d'entreposage des données est la solution incontournable pour répondre aux besoins de l'entreprise EDF. Rappelons que cette technologie a contribué au succès de plusieurs entreprises comme Coca-Cola ou Wallmart. Ce succès a poussé de nombreuses entreprises à adopter cette technologie, que l'on retrouve aujourd'hui dans plusieurs domaines comme le trafic routier [10], la santé [14, 41], l'agriculture [88], la gestion des incendies des forêts [23] ou encore les sciences de la terre [89] pour la modélisation et l'interprétation de données relatives aux tremblements de terre afin d'évaluer et d'améliorer les capacités de prévision.

Dans ce chapitre nous définissons ce qu'est un entrepôt de données (*ED*) et quelles en sont les principales caractéristiques. Nous présentons ensuite leur conception, également appelée cycle de vie. Puis nous nous attachons à décrire la contribution des ontologies dans la conception des entrepôts de données. La dernière section présente l'exploitation des entrepôts de données par la théorie des jeux sur des questions de facturation dynamique.

# 2 Définition et caractéristiques d'un entrepôt de données

Dans son ouvrage de référence «Building the Data Warehouse» [56], *W. Inmon* définit un entrepôt de données comme étant «*une collection de données intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'interrogation et l'analyse*». Cette définition englobe différents termes que nous explicitons ci-dessous :

- *Orientées sujet* : souvent les bases de données organisent les données par application ou par service dans l'entreprise (inventaire, catalogue, etc.). Dans un *ED* ces données sont organisées par sujets (production, ventes, clients, etc.).
- *Intégrées* : les données proviennent de différentes sources, souvent hétérogènes sur les plans syntaxique et sémantique. La phase d'intégration permet d'éliminer les conflits d'hétérogénéités afin d'uniformiser les données avant leur chargement dans l'*ED*. C'est une phase considérée comme critique par *W. Inmon* [56]. On retrouve cette idée de phase critique dans un rapport de *Gartner* [42] qui indique qu'il s'agit de la principale source d'échec des projets d'entreposage.
- *Non volatiles* : toutes les données sont conservées et ne sont accessibles qu'en mode consultation. Les données ne sont ni modifiées ni supprimées afin de préserver les capacités d'analyse.
- *Historisées* : les données sont associées à un repère temporel pour refléter l'activité d'une entreprise dans le temps.
- *Résumées et disponibles pour l'interrogation et l'analyse* : les données doivent être agrégées afin de faciliter leur analyse et de fournir des rapports facilement exploitables aux

décideurs. Les agrégats sont accessibles avec divers outils : requêtes, outils *OLAP* [12], outils de fouilles de données, etc.

Un entrepôt de données est donc une base de données servant de référence unique dans une entreprise. Il est utilisé dans le but de fournir une aide à la prise de décisions *via* des *analyses statistiques* et des *outils de communication* de données (*reporting*).

## 2.1 Architecture d'un entrepôt de données

La figure 2.1 décrit les composantes principales d'un système d'entreposage de données.

- On retrouve en premier les *sources*. Elles proviennent des différentes applications de l'entreprise. Une partie des données qu'elles contiennent va être stockée dans l'entrepôt de données et ne sera donc plus modifiée contrairement aux données de la source. Les sources peuvent prendre de nombreuses formes ce qui contribue à leur hétérogénéité. On retrouve des bases de données, des tableurs, des fichiers *XML*, etc.
- Juste après les sources on trouve l'intégration des données grâce au processus *ETL* [13]. L'*ETL* est un intergiciel qui permet de réaliser des synchronisations massives de données d'une source de données vers une autre. Dans notre cas entre les sources, comme précisées ci-dessus, et l'*ED*. La synchronisation des données correspond à des transformations depuis le modèle de données de la source vers le modèle de données de la cible et d'éventuelles opérations de filtrage. Le rôle de l'*ETL* est critique car les données qu'il charge dans l'*ED* seront consultées pour prendre des décisions par la suite sans qu'elles puissent être modifiées.
- L'*ETL* charge donc les données dans l'*ED* qui, comme précisé dans sa définition, va conserver toutes les données qu'il reçoit. Toutefois chaque utilisateur ou expert n'aura pas besoin de toute l'information contenue dans l'*ED*. C'est pourquoi des *magasins de données* (*datamarts*) sont créés, ils contiennent une partie de l'information et sont destinés à une application spécifique.
- A partir des éléments précédents il devient possible d'exploiter l'*ED*. L'une des façons les plus courantes est d'utiliser un *serveur OLAP* pour accéder à l'*ED*. Son rôle est déterminant dans la mesure où il va traduire les requêtes des utilisateurs en requêtes sur l'*ED* et fournir les résultats à des outils d'aide à la prise de décisions.
- Enfin les *outils de reporting* sont à destination des décideurs. Ils peuvent prendre différentes formes : graphiques, vues multidimensionnelles, rapports, etc.

La figure 2.1 représente également les principales étapes de conception à respecter. Ces étapes sont présentées plus en détails dans la suite de cette section.

12. *OnLine Analytical Processing* : traitement analytique en ligne

13. *Extract, Tansform and Load* : la phase, ou processus, *ETL* regroupe les opérations d'extraction de données des sources, de transformation des données au format du stockage ciblé et de leur chargement dans le stockage ciblé

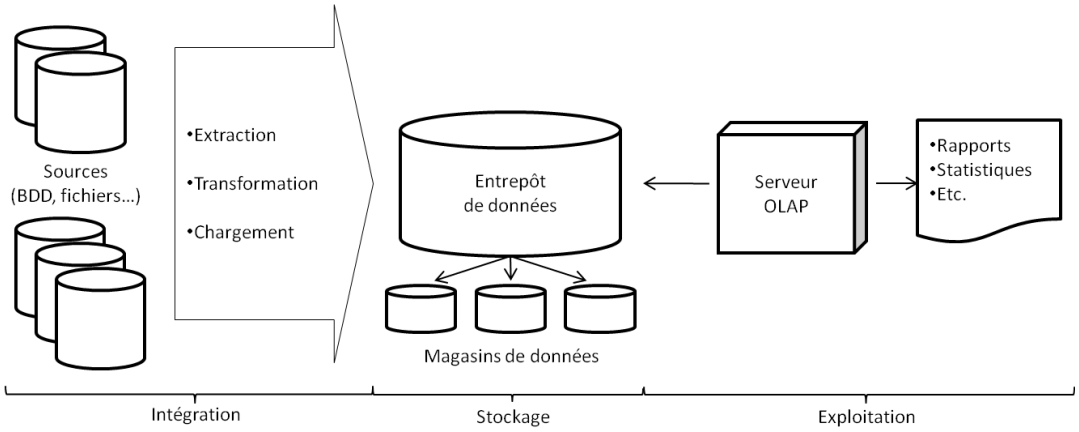


FIGURE 2.1 – Architecture du fonctionnement d'un entrepôt de données

## 2.2 Modélisation multidimensionnelle

Au départ, l'environnement d'analyse en ligne de données *OLAP* est exploité par des applications d'aide à la prise de décisions. Une nouvelle approche a été proposée dans les années 90 par *R. Kimball* [65] : il s'agit de la *modélisation multidimensionnelle*. Cette méthode est maintenant reconnue comme la modélisation la plus adaptée aux besoins d'analyse et de prise de décisions [5].

La modélisation multidimensionnelle, en soi, est une technique de conceptualisation et de visualisation de modèles de données qui offre une structure et une organisation des données propices à l'analyse des données. L'idée principale est d'offrir une vue multidimensionnelle de la donnée au cœur de l'analyse. Le sujet à étudier va être mis en évidence au centre de la modélisation et mis en perspective par les différents axes d'analyse. Ce concept est proche de la manière dont un expert organise son analyse autour de son sujet. Ainsi la modélisation multidimensionnelle est organisée autour de deux éléments :

- Un *fait*, c'est le sujet étudié indiqué précédemment, il présente un intérêt pour l'entreprise (les charges des utilisateurs de $\mathcal{VE}$, les trajets des $\mathcal{VE}$, etc.). C'est toujours un concept central pour la prise de décisions (qui va se charger à ce moment précis, où la charge peut avoir lieu, etc.). Un fait est composé d'attributs, ou mesures, qui correspondent à son thème (pour une charge : l'énergie chargée, le lieu, la date, etc.). Les faits sont ensuite analysés selon différents axes.
- Les *axes*, ou dimensions, sont les contextes d'analyse des faits (le temps, les lieux, les types d'utilisateurs, etc.). Les axes sont formés par des listes d'éléments hiérarchisées tels : années, saisons, mois, jours. Parmi les hiérarchies, on définit celles qui sont complètes lorsqu'elles vérifient la propriété suivante : les objets d'un niveau de la hiérarchie appartiennent à une seule classe d'objet d'un niveau supérieur.

Le noyau d'un *ED* est constitué par son **schéma**, c'est-à-dire le modèle multidimensionnel qu'il possède. Il est dit en étoile (voir figure 2.2) ou en flocon de neige (voir figure 2.3) selon

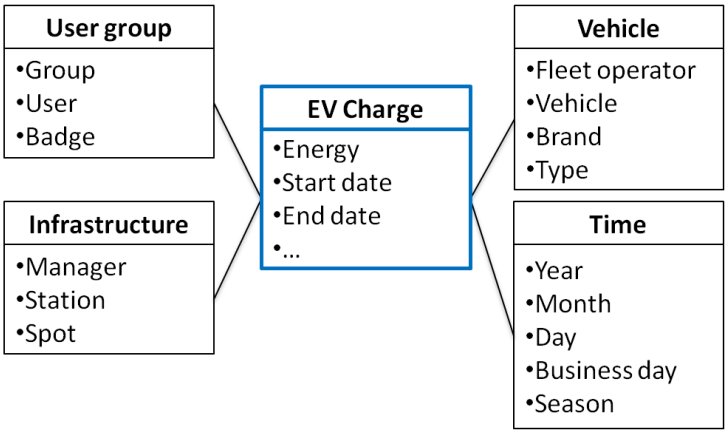


FIGURE 2.2 – Schéma en étoile (en bleu la table des faits)

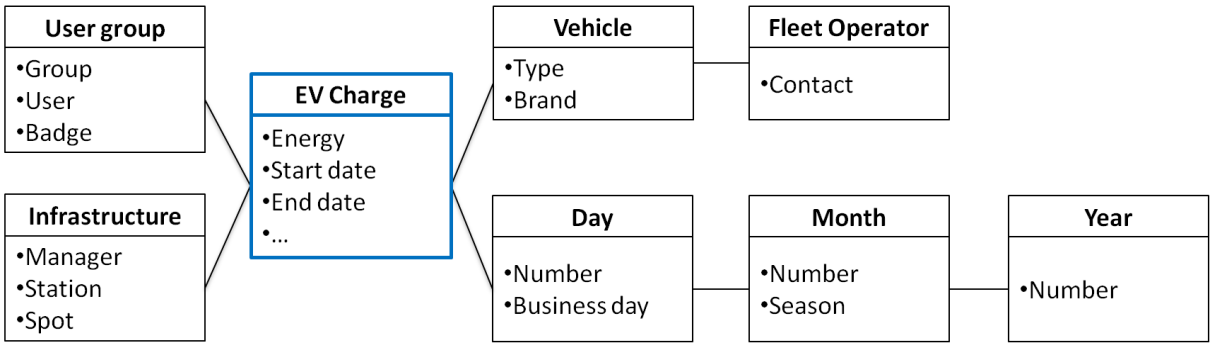


FIGURE 2.3 – Schéma en flocon de neige (en bleu la table des faits)

la nature des dimensions, ou table de dimensions : des listes pour le modèle en étoile et des hiérarchies qui se séparent pour le modèle en flocon.

# 3 Cycle de vie de construction d'un entrepôt de données

Comme tout produit informatique, la conception d'un *ED* a son cycle de vie qui regroupe les phases suivantes [65] :

– Le cycle de vie commence par une phase de *planification*. Comme indiqué plus tôt un *ED* est destiné à apporter des éléments à des analyses menées par les décideurs. Il convient donc de réfléchir à la finalité de l'entrepôt, à son périmètre pour former une sorte de cahier des charges. Cette étape vise à spécifier l'entrepôt en accomplissant les tâches suivantes :

  - Déterminer les buts de l'entrepôt à développer et les décisions qu'il sera amener à aider. Cette réflexion doit permettre de réaliser la modélisation multidimensionnelle en choisissant les faits et les dimensions selon lesquelles les analyser.

  - Évaluer la faisabilité technique (les sources sont-elles disponibles ?) et économique (l'investissement de temps et de moyens est-il rentable ?). Cette étape nécessite de poser

des questions relatives aux sources disponibles, au coût de développement de l'$ETL$, de la maintenance, etc. Et de comparer ces réponses avec les résultats attendus par l'exploitation de l'entrepôt.

- Identifier les utilisateurs et leurs rôles. Dans cette phase le modèle multidimensionnel peut être étoffé, des *datamarts* définis, etc.

– *Conception et implémentation* : dans cette phase le schéma de l'entrepôt de données sera développé : choix du type de schéma, définitions des hiérarchies, des attributs des faits, etc. Il faut également préparer le déploiement en choisissant une plate-forme logicielle, les ressources, etc.

– *Maintenance* : cette phase concerne la mise à jour de l'entrepôt de données avec de nouvelles données et l'optimisation des requêtes, la mise à jour de l'$ETL$ en fonction des nouvelles sources, etc.

La phase de conception à elle seule comporte cinq principales étapes qui forment le **cycle de conception** [48] :

1. l'analyse des besoins,
2. la modélisation conceptuelle,
3. la modélisation logique,
4. le processus d'extraction, de transformation et de chargement ($ETL$),
5. et une phase de conception physique.

Ces étapes sont décrites plus en détails ci-après.

## 3.1 Définition des besoins

L'analyse des besoins est une étape essentielle dans tout projet de développement de logiciel. Elle permet de réduire les risques d'échec d'un projet [46]. La spécification des besoins d'un projet d'entreposage de données permet d'identifier : (1) les différentes fonctions de l'entrepôt, (2) l'ensemble des informations requises et (3) les données qui doivent être accessibles [22]. L'analyse des besoins permet l'identification des besoins des utilisateurs et décideurs dans le but de fournir un modèle répondant aux exigences de l'entreprise [8]. Cette phase passe par trois étapes essentielles :

1. Collecte des besoins : elle s'effectue dans le but de comprendre le domaine à modéliser. Deux principales catégories de méthodes ont ainsi été proposées : une première catégorie orientée sources se basant uniquement sur les sources de données et une deuxième catégorie orientée besoins reconnaissant la nécessité d'inclure les besoins des utilisateurs de l'$ED$ lors de sa conception. Cela en effectuant de manière itérative des interviews, des techniques de réunions participatives et des séances de *brainstorming*. Des méthodes dites mixtes (ou hybrides) ont ensuite été proposées afin de définir le schéma de l'entrepôt à partir des sources et des besoins des utilisateurs.

2. Analyse des besoins : cette étape permet d'analyser les données collectées afin de détecter des besoins conflictuels, contradictoires, complémentaires, etc.
3. Validation des besoins : cette étape permet la validation des modèles initiaux en présence des utilisateurs concernés.

En ce qui nous concerne, nous avons réalisé cette phase en interrogeant les experts d'EDF sur leurs besoins, en complément des échanges réalisés pendant la conception de l'ontologie.

## 3.2 Modélisation conceptuelle

La modélisation conceptuelle consiste à définir une méthode d'élaboration d'un schéma conceptuel multidimensionnel de l'*ED*. L'objectif étant de fournir une représentation abstraite du domaine étudié indépendamment du système de stockage utilisé. Contrairement à la modélisation logique, il n'existe à ce jour aucun consensus sur la modélisation conceptuelle des *ED*. Certains travaux ont été développés pour fournir le même cadre conceptuel que nous trouvons pour les bases de données traditionnelles. Dans [47], les auteurs proposent une méthodologie semi-automatique pour concevoir un modèle conceptuel d'un *ED*, appelé *Dimensional Fact model* (*DF model*). Ce modèle conceptuel est construit à partir de schémas entité-association qui représentent les bases de données sources. Le *DF model* consiste en un ensemble de schémas, appelés schémas de faits, représentés graphiquement sous forme d'un arbre dont les éléments de base sont les faits, attributs, dimensions et hiérarchies.

## 3.3 Modélisation logique

La modélisation logique d'un entrepôt est basée sur le type de schéma choisi. Deux alternatives sont possibles :

1. Un schéma relationnel (schéma en étoile, en flocon de neige ou en constellation) où les données sont stockées dans un système de gestion de bases de données (*SGBD*) relationnel.
2. Un schéma multidimensionnel (cube) où les données sont stockées dans une base de données multi-dimensionnelles.

Dans les premiers projets d'entreposage de données, les concepteurs débutaient directement par le modèle logique, cela impliquait une forte maîtrise par les utilisateurs des modèles informatiques manipulant l'entrepôt final. D'où l'intérêt d'avoir une modélisation conceptuelle qui servira comme support de l'interrogation de l'entrepôt.

### 3.3.1 Modélisation relationnelle

Le premier schéma relationnel adopté par la communauté est le schéma en étoile (*star schema*) [64]. Dans ce type de schéma, les mesures sont représentées par une table de faits et

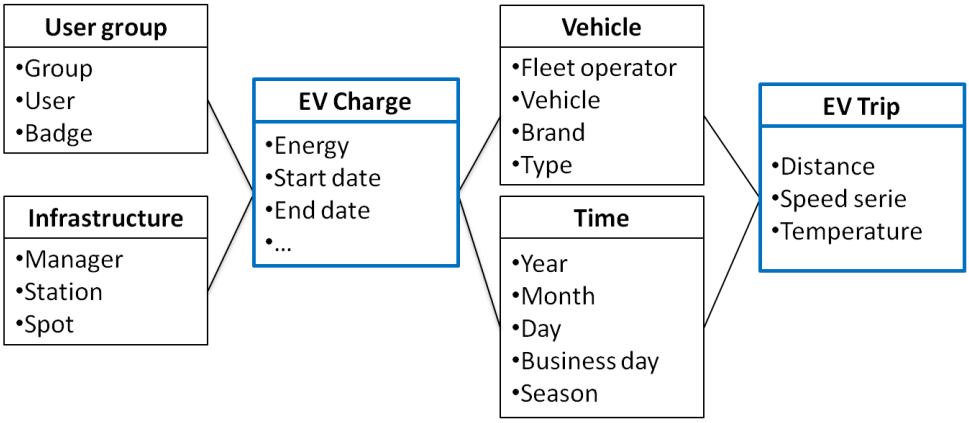


FIGURE 2.4 – Schéma en constellation (en bleu les tables des faits)

chaque dimension par une table de dimension. La table des faits se trouve au milieu de l'étoile et les tables de dimension dans les branches. La table des faits est normalisée et volumineuse. Les tables de dimension sont généralement dé-normalisées afin de réduire le nombre de jointures nécessaires à l'évaluation d'une requête. La figure 2.2 décrit un exemple de schéma en étoile où la table des faits *EV Charge* est liée par des clés étrangères aux tables de dimensions *Time, Infrastructure, Vehicle* et *User group*.

Les requêtes définies sur ce schéma sont appelées requêtes de jointure en étoile. Elles ont les caractéristiques suivantes :

- Toute jointure passe par la table des faits.
- Chaque table de dimension est impliquée dans une opération de jointure à plusieurs prédicats de sélection sur ses attributs descriptifs.

### 3.3.2 Schéma en flocon de neige (*snowflake schema*)

Ce schéma dérive du précédent avec une table centrale et autour d'elle les différentes dimensions. Ces dernières sont décomposées en hiérarchies. Ce schéma a été proposé dans le but de normaliser les tables de dimensions et mettre en évidence la hiérarchie entre les dimensions. Les tables représentant les hiérarchies les plus fines sont directement liées à la table des faits. Celles représentant les autres hiérarchies sont liées entre elles selon leur niveau dans cette hiérarchie. La figure 2.3 décrit un exemple de schéma en flocon de neige.

### 3.3.3 Schéma en constellation

Les schémas en constellation sont utilisés dans des situations nécessitant plusieurs tables de faits. Les tables de faits forment une famille partageant plusieurs relations de dimensions. Les tables de dimensions partagées doivent être exactement les mêmes [64]. La figure 2.4 décrit un exemple d'un schéma en constellation.

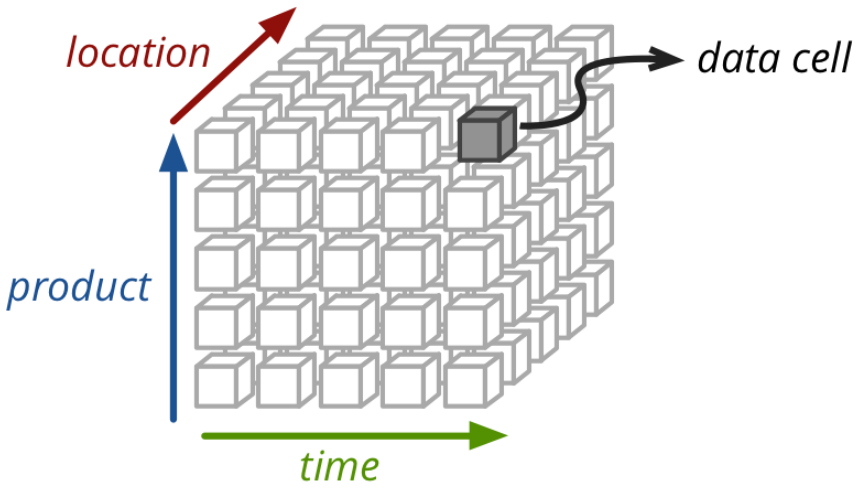


FIGURE 2.5 – Cube de données à 3 dimensions, l'élément *data cell* désigne un enregistrement

## 3.4 Conception Multidimensionnelle

Le modèle multidimensionnel repose sur le concept d'hypercube pour représenter les données. Un hypercube organise les données en une ou plusieurs dimensions. Il est composé de cellules qui représentent les mesures. La figure 2.5 décrit un exemple de cube multidimensionnel qui permet l'analyse des mesures selon les différentes dimensions : *product, location and time*. Toute opération sur le cube est considérée comme une opération *OLAP*. On peut distinguer : le *slice and dice*, *drill down* (le forage vers le bas), *roll up* (forage vers le haut), etc.

## 3.5 Processus ETL

La phase d'extraction, transformation et chargement ($ETL$) est la phase la plus importante du cycle de vie. La qualité de l'entrepôt dépend fortement de celle d'$ETL$ pour éviter d'inclure des données de mauvaise qualité qui généreront de mauvais rapports (phénomène *Garbage In - Garbage Out*). $ETL$ est un processus permettant l'intégration des données provenant de différentes sources hétérogènes. Le processus est responsable de l'intégrité et de l'exactitude des données de l'$ED$ et par extension des décisions prises par l'entreprise avec l'$ED$. Tout d'abord, les données sont extraites à partir des sources de données hétérogènes (bases de données, fichiers plats, $ERP$ [14], documents Web, etc.). Ensuite, elles sont propagées dans une zone de stockage temporaire : *Data Staging Area (DSA)*, où leur transformation, homogénéisation et nettoyage auront lieu. Enfin, les données sont chargées dans l'$ED$ cible. $ETL$ est un processus coûteux en termes de temps et d'argent [137], il consomme jusqu'à 70% des ressources [115] allouées à un projet.

14. *Enterprise Resource Planning* est défini comme étant l'interconnexion et l'intégration de l'ensemble des fonctions de l'entreprise dans un système informatique centralisé

Actuellement, pléthore d'outils $ETL$ a été proposée par la communauté industrielle comme Microsoft, Oracle, IBM [15], Talend, et la communauté académique [138]. Cependant, ces outils suivent différentes techniques de modélisation et utilisent différents langages de programmation.

Les algorithmes $ETL$ sont souvent définis sur les modèles physiques de chaque source. Cela augmente le coût de développement [13]. Récemment, certains travaux ont proposé de redéfinir les algorithmes $ETL$ au niveau conceptuel afin de cacher les aspects liés à l'implémentation [13].

### 3.6 Modélisation physique

Durant la phase physique, l'administrateur doit sélectionner des structures d'optimisation comme les index ou les vues matérialisées pour optimiser les accès à l'$ED$ [72].

Dans la littérature, il existe plusieurs types de structures d'optimisation pour la conception physique. Le choix des structures pertinentes et adéquates pour optimiser une charge de requêtes est un problème complexe. Il nécessite une certaine expertise. Ce choix peut se faire manuellement en se basant sur l'expérience de l'administrateur et les caractéristiques des requêtes. Mais on peut aussi se servir des outils proposés par les $SGBD$ commerciaux et non commerciaux comme SQL Access Advisor [71], DB2 Design Advisor [148], et Parinda pour le $SGBD$ PostgreSQL [80].

## 4 Ontologies dans le monde des entrepôts de données

Avec l'émergence des ontologies dans plusieurs domaines et surtout leur similarité avec les modèles conceptuels, la communauté de recherche autour des $ED$ s'est intéressée activement à leur connexion au cycle de vie. En examinant la littérature, nous pouvons constater que les ontologies ont été utilisées pour l'ensemble des phases de cycle de vie mais d'une manière isolée (voir figure 2.6).

### 4.1 Ontologie au niveau source de données

Les ontologies ont été utilisées pour préparer les sources de données afin d'expliciter leur sémantique et en conséquence faciliter leur intégration. Rappelons que l'hétérogénéité des sources de données est à l'origine de la complexité de la tâche d'intégration. Cette hétérogénéité peut être de deux natures : structurelle (ou syntaxique) ou sémantique.

15. International Business Machines

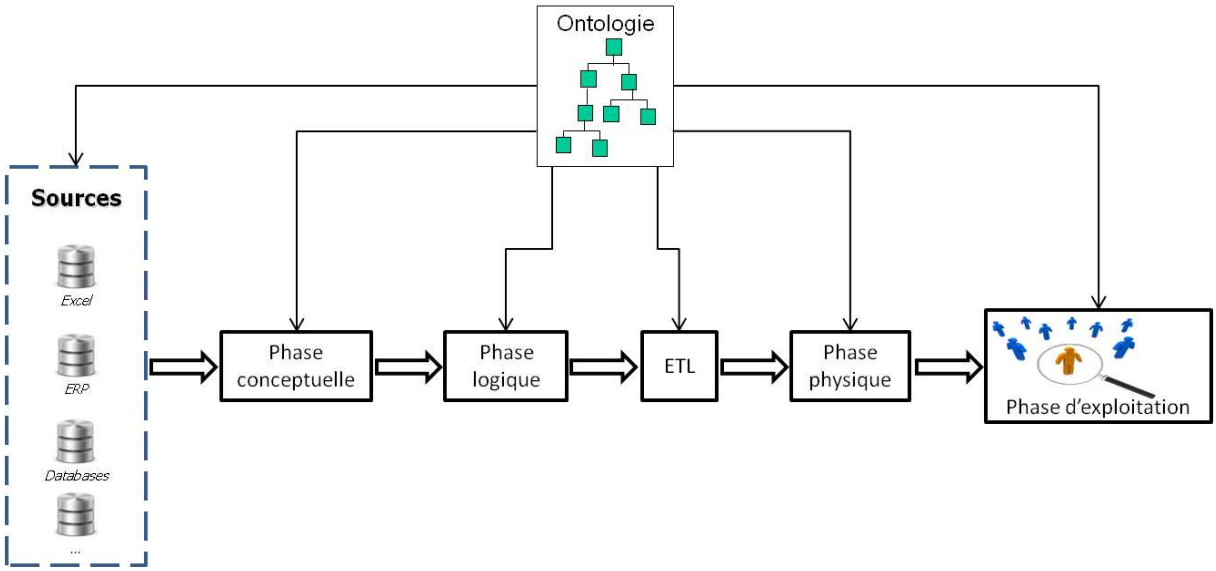


FIGURE 2.6 – Projection de l'ontologie sur le cycle de vie d'un entrepôt de données

### 4.1.1 Hétérogénéité structurelle

L'hétérogénéité structurelle provient du fait que les sources de données peuvent avoir différentes structures ou différents formats de stockage. Les différences structurelles peuvent être regroupées dans différentes catégories :

1. Choix du type de données : ces conflits se posent lorsqu'on utilise des types de données différents pour la même information. Par exemple, dans le domaine des transactions commerciales, la quantité d'un produit est représentée par un réel dans une source *S1* et par une chaîne de caractères dans une autre source *S2*.
2. Choix du nombre de constructeurs : ces conflits se présentent lorsque le nombre de constructeurs modélisant une information est différent d'une source à une autre. Par exemple, l'attribut nom d'un client est modélisé par un seul attribut servant à stocker le nom et le prénom d'un client dans une source *S1*, alors que deux attributs sont utilisés dans une autre source *S2*.
3. Choix des informations représentées : ces conflits se posent lorsqu'une information est représentée dans des sources alors qu'elle ne l'est pas dans d'autres. Par exemple, l'adresse d'un client n'est pas connue pour tous les clients d'une source *S1*, alors que c'est une donnée obligatoire dans une source *S2*.

### 4.1.2 Hétérogénéité sémantique

L'hétérogénéité sémantique représente une problématique plus difficile à gérer. Elle provient du fait que les sources sont conçues par différents concepteurs qui ont des objectifs applicatifs différents et ne partagent donc pas forcément la même sémantique des concepts. Les conflits

sémantiques peuvent être de différentes natures. *Goh et al.* [44] distinguent les trois types de conflits sémantiques suivants :

1. Les *conflits de noms* se produisent lorsqu'on utilise soit des noms différents pour le même concept ou propriété (synonyme), ou plus rarement des noms identiques pour des concepts différents (homonyme). Par exemple : si l'on trouve le concept *Produit* dans la source *S1* et *Article* dans la source *S2*, alors que les deux concepts portent le même sens dans les deux sources. Dans un autre cas, on retrouve l'attribut *Prix* dans les deux sources, mais qui signifie le prix de vente d'un produit dans la source *S1*, et le prix de production d'un produit dans la source *S2*.
2. Les *conflits de contexte* se produisent lorsque des concepts semblent avoir la même signification mais ils sont évalués dans différents contextes. Par exemple, la propriété *Prix* ne s'applique que pour les produits neufs dans la source *S1*, alors qu'elle est appliquée pour tous les produits dans la source *S2*.
3. Les *conflits de mesures* ou *de valeurs* se trouvent dans le cas où des unités de mesure différentes ont été utilisées pour mesurer certaines propriétés de certains concepts. Par exemple, la valeur de l'attribut *Prix* d'un produit est calculée en *dinars* dans la source *S1* et en *euros* dans la source *S2*.

Certains travaux sont allés plus loin en intégrant l'ontologie dans le système de gestion de base de données, d'où la naissance d'un nouveau type de bases de données, appelé base de données à base ontologique (*BDBO*). Une *BDBO* est «*une source de données qui contient des ontologies, un ensemble de données et des liens entre ces données et les éléments ontologiques*» [99]. Une *BDBO* possède donc deux caractéristiques principales :

- Les ontologies et les données sont représentées dans une unique base de données, de ce fait elles peuvent faire l'objet des mêmes traitements.
- Chaque donnée est associée à un élément ontologique qui la définit.

On peut citer *KAON* [18], *OntoDB* [58] ou encore Oracle Semantic [91] comme solution de *BDBO*.

Une autre utilisation des ontologies au niveau des sources est l'annotation. L'annotation sémantique de données (des images, des textes, des éléments de bases de données, etc.) consiste à associer une donnée avec des éléments sémantiques d'une ontologie. Cette démarche permet d'enrichir les données ou de disposer davantage de ressources, ceci afin de permettre aux applications d'apporter des réponses plus pertinentes aux utilisateurs.

Pour démontrer l'intérêt des annotations on peut citer les travaux réalisés sur les annotations sémantiques d'images. L'objectif de ces travaux est de fournir des capacités accrues d'interrogation aux utilisateurs, autrement qu'en cherchant dans le nom des fichiers. Une approche populaire consiste à chercher dans le texte «autour» de l'image. En effet, sur internet la plupart des photos sont présentées avec le contexte décrit textuellement (lieu, date, personnes présentent, etc.). Le *framework* SIA [16] [68], par exemple, se focalise sur l'image en calculant des

16. Semantic Image Annotation

indicateurs : distribution des couleurs sur l'image, couleurs présentes, types de textures, etc. La première étape est d'analyser puis d'annoter les images par ces indicateurs. La deuxième étape consiste à traduire les requêtes des utilisateurs en indicateurs. A partir de ces deux étapes le *framework* propose des listes de résultats aux requêtes par ordre de similarité entre les indicateurs de la requête et ceux des images.

Cet exemple montre que l'annotation de données aussi complexe qu'une image permet de déployer des services pointus tel la recherche d'image en langage courant.

Réciproquement l'annotation sémantique peut contribuer à enrichir une ontologie en découvrant de nouveaux concepts. Cette approche est présentée dans [7]. Dans leurs travaux ils exploitent un *noyau d'ontologie* lié à des textes dont l'analyse sémantique permet d'enrichir l'ontologie.

## 4.2 Projection des ontologie sur les besoins

EDF est une entreprise globale, qui possède différentes branches, comme la branche commerce, la branche R&D, etc. Ces branches sont divisées, par exemple la R&D est composée de départements spécialisés. Certaines de ses branches sont même internationales. La diversité des experts qui composent ces divisions est à l'origine d'une source d'hétérogénéité importante dès lors qu'il faut définir des besoins sur un même sujet. Chaque département va utiliser son propre vocabulaire pour décrire ses besoins. Au delà de leurs noms, les concepts employés ne vont pas non plus être exactement équivalent à ceux d'un autre département.

Grâce à l'ontologie, qui va permettre d'avoir une référence commune, il devient possible d'unifier ces besoins. Cette problématique de description des besoins au niveau sémantique est connue. Les travaux de *I. Boukhari* [17], menés au LIAS, ont permis de projeter une ontologie sur la phase de définition des besoins du cycle de vie de l'*ED*.

## 4.3 Projection de l'ontologie sur la phase conceptuelle

Les ontologies permettent de progresser vers le modèle conceptuel d'une base de données (*BDD*) ou d'un *ED*. Plusieurs travaux ont proposé des méthodes exploitant une ontologie pour la création du modèle conceptuel, on peut par exemple citer :

- [129] qui propose une méthode et un outil de conception de *BDD* reposant sur une ontologie linguistique. Cette méthode suggère des concepts de l'ontologie à partir d'un cahier des charges exprimé en langage naturel. On retrouve toute la puissance exprimée dans l'annotation des éléments du domaine par une ontologie pour servir d'interface entre le langage naturel et des concepts formels. Les auteurs proposent ensuite des contraintes d'intégrité pour valider le modèle conceptuel généré.
- [35] qui propose une méthode de conception de *BDD* : Spécialisation, Importation Sélective et Représentation des Ontologies (*SISRO*). Il s'agit de sélectionner un sous-ensemble

de concepts d'une ontologie pour former le modèle conceptuel.
– [82] qui utilise l'ontologie pour aller au delà du modèle conceptuel et utiliser une ontologie sur le modèle physique, comme par exemple pour les vues matérialisées.

## 4.4 Projection de l'ontologie sur ETL

La phase de modélisation $ETL$ a été introduite dans le cycle de conception des $ED$ de manière arbitraire pour répondre à la problématique d'intégration et de chargement des données dans un cycle de conception qui ne comprenait que les deux principales phases de modélisation logique et physique. Par conséquent, nous remarquons trois principales catégories d'approches $ETL$ qui se dégagent selon le niveau d'abstraction : les approches orientées niveau physique, les approches orientées niveau logique et les approches orientées niveau conceptuel. Ces dernières tentent d'élever certaines tâches de la phase $ETL$ au niveau conceptuel. L'objectif de définir la phase $ETL$ à un niveau d'abstraction plus élevé peut être de : fournir une documentation des correspondances entre les sources et l'$ED$ dès le début du projet d'entreposage [118], faciliter l'interrogation des cubes multidimensionnels [126], faciliter la maintenance de l'$ED$ [132], ou automatiser le processus $ETL$ [119].

De nombreux travaux proposent d'exploiter les ontologies de domaine comme modèles conceptuels. Différents éléments peuvent être définis au niveau ontologique : le schéma global, les schémas des sources, les *mappings*, ou le processus $ETL$. Certains travaux proposent d'exploiter la sémantique d'une ontologie pour décrire le schéma global et/ou les schémas des sources, dans le but de faciliter le processus $ETL$. Par exemple, *Bergamaschi et al.* proposent dans [11] une approche $ETL$ exploitant un thésaurus ontologique pour faciliter l'alimentation de l'$ED$. Les schémas des sources et de l'$ED$ sont préalablement définis au niveau logique relationnel. Un thésaurus décrivant les schémas des sources est défini et exploité pour générer les *mappings* entre les schémas des sources et le schéma de l'$ED$. Ces *mappings* sont basés sur des mesures de similarité sémantiques. *Skoutas et al.* proposent dans [119] une approche de conception $ETL$ basée sur une ontologie $OWL$ et des annotations sémantiques, afin de spécifier formellement et explicitement la sémantique des schémas sources et cible de l'$ED$. Le but de l'approche est d'automatiser le processus $ETL$. Les auteurs étendent l'approche dans [120] en considérant les données structurées (relationnelles) et semi-structurées représentées en $XML$. Les auteurs étendent à nouveau cette dernière approche dans [117, 116] en proposant d'exploiter une ontologie linguistique afin de fournir une description textuelle (en langue naturelle) des résultats de la phase de conception $ETL$, à savoir les annotations des sources de données et le scénario $ETL$ généré. Cette description textuelle permet de faciliter la validation de la phase $ETL$ par les utilisateurs et les concepteurs. *Romero et al.* proposent dans [107] une approche ontologique semi-automatique permettant la génération d'un modèle conceptuel multidimensionnel et la représentation conceptuelle des processus $ETL$. Enfin, *Nebot et al.* proposent dans [85, 86] une approche ontologique semi-automatique permettant la génération d'un $ED$ peuplé avec des données du Web qui sont annotées par une ontologie.

## 4.5 Projection de l'ontologie sur la phase logique

L'émergence des *BDBO*, permettant le stockage des ontologies au sein d'une *BDD*, a fait apparaître de nombreux modèles de stockages logiques et physiques. Contrairement aux *BDD* conventionnelles (où le modèle logique est stocké habituellement selon une représentation relationnelle), dans une *BDBO* une variété de modèles de stockages est utilisée pour stocker deux niveaux de modélisation : le niveau du *modèle de l'ontologie* et le niveau des *instances ontologiques*.

Une première approche utilisée pour la représentation des données dans une *BDD* relationnelle est l'approche *table universelle* qui stocke toutes les propriétés dans une seule table universelle. Cette représentation comporte de nombreux inconvénients (grand nombre de colonnes et plusieurs colonnes représentant des valeurs nulles). Trois autres approches plus répandues ont été suivies pour la représentation des ontologies dans une *BDBO* [4, 36] : l'approche *verticale*, l'approche *binaire* et l'approche *horizontale*.

L'approche **verticale** représente toutes les données (schéma et instances ontologiques) par une unique table de triplets composée des trois colonnes (Sujet, Prédicat, Objet). Ces trois colonnes représentent respectivement l'identifiant d'un élément de l'ontologie, un prédicat et la valeur du prédicat. Cette représentation facilite l'insertion de nouveaux triplets. La mise à jour d'une information peut être plus difficile si elle nécessite l'accès à plusieurs triplets, ce qui implique plusieurs auto-jointures. Plusieurs systèmes de *BDBO* utilisent cette représentation, tels que *RStar* [79], *RDFSuite* [6], *KAON* [141] et *Sesame* [21]. Une étude comparative entre ces différents systèmes est présentée dans [1].

Dans une approche **binaire**, les classes et les propriétés auxquelles appartiennent les instances ontologiques sont stockées dans des tables de structures différentes. La table de chaque propriété est constituée de deux colonnes : *id* pour l'identifiant de la propriété, et *value* pour la valeur de la propriété. Cette approche n'est généralement pas adaptée aux larges ontologies et souffre de l'altération du schéma lorsque l'ontologie évolue. *RDFSuite* [6], *Genea* [69], *IBMSOR* [77] et *DLDB − OWL* [92] sont des exemples de *BDBO* qui utilisent cette représentation. *RDFSuite* supporte les deux représentations verticale et binaire. Cette approche possède trois variantes :

1. L'*approche table unique* qui utilise une seule table pour stocker toutes les classes de l'ontologie et une table pour chaque propriété.
2. L'*approche ISA* qui consiste à utiliser les caractéristiques des *SGBD* relationnels-objets pour représenter l'héritage des classes et des propriétés en utilisant la définition *subtable*. Chaque table d'une classe contient une seule colonne représentant les identifiants des instances de cette classe.
3. L'approche *NOISA* qui consiste à ne pas utiliser l'héritage des tables. Les tables de classes et de propriétés sont définies séparément sans être mises en relation.

L'approche **horizontale** consiste à associer à chaque classe de l'ontologie une table d'ins-

tances ayant une colonne pour chaque propriété associée à une valeur pour au moins une instance de la classe. Les *BDBO OntoDB* [54], *OntoDB2* [36] et *OntoMS* [93] utilisent cette représentation. L'héritage des tables est représenté si le *SGBD* utilisé le permet. Quelques systèmes utilisent une approche *hybride* comme : *Jena2* [146] qui combine la représentation verticale et binaire (en représentant les tables triplets et en regroupant quelques attributs) ou *DLDB* [92] qui combine l'approche binaire et horizontale.

Selon le type du schéma ontologique matérialisé dans la *BDBO*, trois principales architectures de *BDBO* sont distinguées [36] : architecture de *type I*, *type II* et *type III*.

Dans les *BDBO* de **type I**, comme dans les *BDD* classiques, l'ontologie et les données à base ontologique (*DBO*) sont stockées dans un même schéma. Il n'y pas de séparation entre l'ontologie et les données. Cette architecture utilise généralement la représentation verticale où toutes les informations de l'ontologie sont représentées en utilisant un seul schéma composé d'une seule table de triplets ou d'une table verticale. *Jena2* [146] utilise ce type d'architecture.

Dans une architecture de **type II**, l'ontologie et les données associées sont stockées dans deux schémas différents : un schéma pour les *DBO*, et un autre pour l'ontologie. Ce dernier est spécifique au formalisme d'ontologie supportée. Les systèmes *RDFSuite* [6] et *SOR* [77] utilisent cette architecture.

L'architecture de **type III** étend celle de type II en définissant un schéma supplémentaire appelé *méta-schéma*. L'ontologie constitue ainsi une instance de ce méta-schéma. La présence du méta-schéma offre une flexibilité pour l'ontologie et permet : (i) l'évolution du formalisme ontologique utilisé, (ii) un accès générique aux ontologies ainsi qu'aux données et (iii) le stockage de différents constructeurs de formalismes d'ontologies (*OWL*, *DAML + OIL*, *PLIB*, etc). Les systèmes *OntoDB* [54] et son extension *OntoDB2* [36], développés au laboratoire LIAS, présentent des *BDBO* de cette architecture.

## 4.6 Projection de l'ontologie sur la phase physique

La conception physique a eu un grand intérêt dans le contexte des *ED*, où une large panoplie de structures d'optimisation (les vues matérialisées, les index de jointure binaire, la compression, le partitionnement, le traitement parallèle, etc.) a été étudiée et supportée par les *SGBD* commerciaux et académiques. La synergie entre les ontologies et les structures d'optimisation citées a suscité un intérêt particulier ces dernières années.

La conception physique est devenue une étape primordiale dans le cycle de vie de conception des bases de données avancées. Durant cette phase, des structures d'optimisation de requêtes sont sélectionnées. Les caractéristiques des *BDBO* rendent plus complexe cette conception. Cela est dû à leur diversité qui porte sur :

1. Des formalismes supportés : chaque *BDBO* utilise un formalisme particulier pour définir ses ontologies (*OWL*, *PLIB* ou *FLIGHT*),

2. Des modèles de stockage : une variété de modèles de stockage (représentation horizontale, spécifique, verticale, etc.) sont utilisés pour les *BDBO*,

3. Des architectures des *SGBD* cibles supportant ces bases.

Trois types d'architecture existent : (a) l'architecture «deux quarts» qui est celle des *BDD* traditionnelles, où les ontologies et les données sont stockées ensemble. (b) Le second type d'architecture, qualifié de «trois quarts», sépare les ontologies des instances ontologiques. (c) La dernière architecture, dite «quatre quarts», ajoute à la précédente une partie appelée méta-schéma.

Souvent, la conception physique est guidée par des modèles de coût mathématiques estimant la performance des requêtes. Elle sert surtout de support aux algorithmes de sélection de structures d'optimisation. Dans la thèse de *B. Mbaiossoum* [82], développée au laboratoire LIAS, des modèles de coûts mathématiques ont été développés pour estimer le coût de chargement des données ontologiques et de l'ontologie, le temps de réponse des requêtes (en termes de nombre d'entrées-sorties entre le disque et la mémoire) pour chaque type de *BDBO*.

# 5 Entrepôt de données pour la théorie des jeux

## 5.1 Contexte économique : EDF et la mobilité électrique

Les modèles de facturation conçus et utilisés par EDF possèdent des caractéristiques bien précises. La consommation électrique en dehors de la mobilité électrique ($ME$) est *statique* : elle est reliée à un endroit géographique précis. L'échelle peut varier d'un logement à un site industriel [17] ou à une commune mais tous ces points de consommation sont géolocalisables. La facturation est basée sur des relevés manuels des compteurs de ces lieux de consommation. L'historique des relevés est, quant à lui, archiver et étudier afin de proposer aux consommateurs des tarifs adaptés à leurs besoins.

Comme précisé dans l'introduction, la $ME$ est un nouveau marché qui apportent de nombreux changements. En premier lieu, les clients sont mobiles. Ce simple fait remet en question les modèles de facturation déjà établis car ces derniers se basent sur des points de distribution. Ensuite le comportement des utilisateurs n'est pas un phénomène aussi continu et localisé que la consommation d'un foyer par exemple. Pour un $\mathcal{VE}$ la consommation va être brusque (l'appel en puissance correspond à minima à celui d'un four électrique ménager à pleine puissance pour une charge lente), durer un moment avant disparaître et cela sur un point précis du réseau.

Dans ces conditions et avec un entrepôt de données il devient possible d'étudier le comportement de l'utilisateur pour détecter les zones où il se charge et essayer d'apprendre ses habitudes.

17. http://medias.edf.com/communiques-de-presse/tous-les-communiques-de-presse/communiques-2009/edf-energies-nouvelles-signe-un-contrat-de-vente-40177.html

Seulement nous ne sommes plus dans un contexte où l'approvisionnement en électricité de n'importe quel point du réseau de distribution est garanti à n'importe quel moment . Maintenir le réseau de distribution existant a un coût, le développer pour supporter l'usage croissant de l'énergie a un coût, produire de l'électricité est de plus en plus cher, etc. Ces derniers hivers, un appel de puissance électrique trop important sur le réseau de distribution a eu pour conséquences de priver des régions entières d'électricité. Ces appels trop important sont appelés des *pics de consommation*, ils correspondent à une demande plus forte que l'offre d'électricité sur une portion du réseau. Les pics de ces dernières années ont eu lieu pendant les vagues de froid, en partie à cause de l'utilisation de chauffages électrique, ou pendant les intempéries quand certaines lignes d'approvisionnement sont coupées et que celles qui restent sont insuffisantes.

Les pics de consommation sont donc caractérisés par leur aspect dynamique et par la difficulté de les prévoir longtemps avant leurs apparitions. Si l'on rajoute la consommation des $\mathcal{VE}$ à venir à la consommation actuelle, on se dirige nécessairement vers une augmentation des phénomènes de pics.

Étant donné que le domaine est naissant, la mise en place d'un système de facturation adéquates est un réel besoin pour EDF. Ce besoin cristallise la nécessité d'exploiter les mines de données et de processus d'EDF. Une solution envisageable est de passer par un système de facturation dynamique basé sur la connaissances et l'historique des habitudes des consommateurs contenues dans un *ED*. Une approche possible est de recourir à la théorie des jeux en décrivant les processus métiers nécessaires à sa mise en place. La section ci-dessous présente les concepts fondamentaux de cette théorie.

## 5.2 Concepts fondamentaux de la théorie des jeux

La théorie des jeux a permis de formaliser des situations très diverses depuis le routage dynamique des *IP* [18] sur internet [135] jusqu'à l'équilibre des arsenaux nucléaires entre deux pays [104]. La définition de *Myerson* [84] est communément admise : «*La théorie des jeux est un ensemble d'outils pour analyser les situations dans lesquelles ce qu'il est optimal de faire pour un agent (personne physique, entreprise, animal...) dépend des anticipations qu'il forme sur ce qu'un ou plusieurs autres agents vont faire. L'objectif de la théorie des jeux est de modéliser ces situations, de déterminer une stratégie optimale pour chacun des agents, de prédire l'équilibre du jeu et de trouver comment aboutir à une situation optimale.*»

Cette définition met en avant plusieurs éléments qui sont des caractéristiques des *agents*, également appelés *joueurs*.

- Les *anticipations* : il s'agit d'évaluer la connaissance qu'a un joueur du jeu et des autres joueurs.
- Ce qu'il est *optimal de faire* : les joueurs ont leurs propres objectifs. Leurs actions ont

18. Internet Protocol, c'est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque appareil connecté à internet.

pour but de se rapprocher de leurs objectifs. Les résultats obtenus sont appelés les *gains*.
- Les *stratégies* : chaque joueur, en fonction de ses connaissances, doit choisir sa façon de jouer. Ces choix forment l'ensemble des stratégies d'un joueur.

Les joueurs évoluent ensuite dans un jeu, suivant le type de jeu les outils disponibles ne sont pas les mêmes [25], voici des éléments permettant de classifier les jeux :
- Les jeux *coopératifs* recherchent le bénéfice pour tous les joueurs (création d'une norme).
- Les jeux *à somme nulle* sont des jeux où la somme des gains des joueurs est nulle, ce qui est gagné par l'un est perdu par l'autre (échecs ou poker).
- Les jeux *finis* quand les stratégies des joueurs sont connues et en nombre fini.
- Les jeux *simultanés*, par opposition aux jeux *séquentiels* où les joueurs jouent à leur tour.
- etc.

Suivant les cas, les jeux peuvent appartenir à plusieurs groupes. Notre cas pourrait être un jeu coopératif (on vise le bénéfice de tous), séquentiel (les joueurs vont jouer tour à tour) et fini (les offres d'EDF sont connues des joueurs et EDF connait les réactions possibles des utilisateurs).

## 5.3 Cas d'étude EDF

Pour positionner notre utilisation de la théorie des jeux, voici quelques éléments qu'il nous faut préciser. Ils seront repris en détails dans le chapitre 6 relatif à l'exploitation de notre solution pour EDF. Pour exploiter le formalisme de la théorie des jeux il convient de généraliser notre problématique. L'électricité est la ressource au centre du problème. On peut considérer EDF comme un des joueurs de notre «jeu», ce joueur produit la ressource, doit en assurer la disponibilité et ses gains vont être liés à la demande de la ressource par les autres joueurs. Toutefois dans certains cas les demandes de la ressource vont avoir un impact négatif sur le joueur EDF.

Les autres joueurs sont les consommateurs, ou des groupes de consommateurs, avec leurs habitudes et leurs besoins spécifiques.

## 5.4 Alimentation de la théorie des jeux par un entrepôt de données

Le rapprochement de la théorie de jeux et des *ED* n'est pas quelque chose de nouveau. On retrouve cette idée dans [144] : «*un développement théorique récent, qui promet un potentiel important est la combinaison de fouilles de données dynamiques avec la théorie des jeux. Alors que la fouille de données analyse le comportement des agents en un contexte réel via l'analyse des données qu'ils génèrent, la théorie des jeux essaye d'expliquer théoriquement le comportement de tels agents*».

C'est exactement ce que nous souhaitons mettre œuvre. Les agents ici sont les clients d'EDF, l'interaction d'EDF avec ses clients rentre parfaitement dans le cadre décrit par *Kobielus* [67] (*Big Data Evangelist d'IBM*) où chaque joueur, EDF compris, essaye d'orienter le comporte-

ment du joueur opposé.

EDF souhaite mettre à profit sa mine de données client pour connaître les habitudes de ses clients. Ensuite EDF souhaite émettre des offres pour éviter les pics de consommation précédemment décrits. Ces offres vont éventuellement créer des changements momentanés dans les habitudes des clients. Ces changements seront basés sur les objectifs des clients (charge moins chère sur le moment, réduction de la facture sur le long terme, démarche écologique, etc.). Les changements génèreront les mêmes données que les comportements classiques, à ceci près qu'il sera possible de les relier aux offres émises. Les *ED*, qui vont tout stocker, sont donc un instrument idéal pour étudier les réactions des clients, comprendre leurs stratégies pour, au final, proposer des offres en adéquation avec les besoins d'EDF.

# 6 Bilan

Les *ED* sont aujourd'hui des outils connus et reconnus tant par les industriels que par la recherche académique. Ils sont activement utilisés et font l'objet de nombreuses études. En effet, de leur bon fonctionnement peut dépendre le succès d'une entreprise ou d'un service.

Dans ce chapitre nous avons décrit l'ensemble des phases reconnues du cycle de vie de conception des *ED* : l'analyse des besoins, la conception logique, celle de l'*ETL* et la conception physique. Nous avons vu que les ontologies sont de plus en plus utilisées à plusieurs niveaux : dans les différentes phases de conception et dans la construction ainsi que dans l'exploitation des *ED*.

On remarque que dans toutes les phases des hypothèses sont sous-jacentes aux différents travaux. Les sources sont supposées connues, que ce soit leur existence ou leurs modèles de données. Les questions auxquelles l'*ED* doit répondre sont identifiées au début de la démarche de création. Les analyses à mener ainsi que leurs destinataires sont toujours supposés connus, ce sont même dans certains cas les commanditaires des travaux. Enfin lorsque des ontologies sont mêlées aux *ED*, l'hypothèse que soit l'un soit l'autre existe est toujours présente.

Pour adapter ces techniques de pointe à notre cas d'étude, qui n'est ni aussi bien connu ni aussi stable et où de nouvelles analyses et applications ne manqueront pas d'arriver, il faudra fournir des solutions innovantes. Enfin nous ne disposons pas d'ontologie de domaine ni de l'*ED*, ce qui ne satisfait aucune des hypothèses prises par les travaux présents dans la littérature.

La partie suivante détaille nos contributions pour relever les défis posés par des hypothèses d'existences fortes.

# Deuxième partie

# Contributions

Chapitre 3

# Construction d'une ontologie modulaire

## Sommaire

# 1 Introduction

Pour reprendre le bilan du chapitre 1 : la mobilité électrique ($ME$) est un domaine technique, il contient des informations sur les infrastructures et leurs équipements, les véhicules et leurs caractéristiques ou encore les données. Pour formaliser un tel domaine les *ontologies de domaine* sont toutes indiquées. La connaissance de ce domaine par les experts nous a permis d'identifier des concepts *primitifs* mais aussi la composition de certains concepts primitifs pour former des concepts *définis* indispensables à la description exhaustive de la $ME$. C'est pourquoi nous nous sommes placés dans le cadre des *ontologies conceptuelles non-canoniques*.

Parmi les formalismes disponibles nous avons choisi *PLIB* pour la précision des définitions des concepts et des propriétés. D'autre part ce formalisme, grâce à l'unicité des concepts, supporte l'utilisation de plusieurs langues ce qui est un facteur important pour l'aspect international de la $ME$ et de l'ontologie.

L'état de l'art propose de nombreux usages des ontologies (chapitre 2) mais ces travaux se fondent sur des hypothèses fortes qui sont, pour rappel, l'existence de l'ontologie, la stabilité du domaine ou encore un nombre limité d'acteurs. Dans ce chapitre nous nous sommes attachés à concevoir une nouvelle méthode de construction d'ontologies basée sur des hypothèses moins fortes.

Notre **hypothèse de base** est la suivante : il est possible de disposer d'une ontologie globale, même si cette dernière doit être très abstraite. Il est facile de définir une telle ontologie grâce aux experts sur un domaine technique, tel que le notre. Leurs expertises permettent de distinguer les sous-domaines à traiter, car ils possèdent une connaissance précise des différents champs d'application, des différents métiers et des interactions entre ces derniers. Ce sont ces sous-domaines qui vont former l'ontologie globale et valider notre hypothèse. A partir de cette ontologie abstraite nous proposons notre méthode de construction adaptée aux contraintes, celles-ci peuvent se résumer par l'impossibilité de vérifier les hypothèses classiques précédemment citées.

Dans le chapitre précédent nous avons vu l'intérêt de disposer d'un entrepôt de données ($ED$) pour analyser le domaine de la $ME$. Comme le montre les travaux cités dans l'état de l'art, la mise en place d'un $ED$ peut être facilitée par l'exploitation d'une ontologie sur le domaine concerné, principalement pour créer les différents modèles (conceptuel, physique et logique).

L'utilisation d'une ontologie permet de persister les besoins des utilisateurs ou de définir un langage commun pour décrire le domaine. Le principal inconvénient de ces méthodes est qu'elles supposent l'existence de l'ontologie à utiliser or il n'existe, aujourd'hui, pas d'ontologie sur la $ME$.

Nous sommes dans un cas classique où la décision de créer une ontologie implique de tenir compte du retour sur investissement envisageable. D'un côté, une ontologie permet de mettre en place des techniques de pointe sur : la conception de systèmes d'informations ($SI$), la communication de données, la définition formelle et consensuelle du domaine, etc. La formalisation d'un domaine permet aussi de créer des outils génériques et plus faciles à maintenir. De l'autre côté,

la création d'une ontologie est un processus long et coûteux : il faut définir les concepts puis les partager afin, à terme, d'atteindre un consensus global de tous les acteurs. A ces considérations s'ajoutent les besoins opérationnels : il faut pouvoir mettre la solution en œuvre rapidement avec des ressources limitées. L'analyse des méthodes de construction d'ontologies, présentées dans ce chapitre, montre qu'il n'y a pas de méthodes satisfaisantes à l'égard des contraintes opérationnelles.

Afin de diminuer l'investissement pour disposer rapidement d'une ontologie opérationnelle nous avons proposé une nouvelle méthode de création d'ontologie.

La première section de ce chapitre propose une brève synthèse de l'état de l'art sur les ontologies. La deuxième section présente notre contribution avec une méthode de construction incrémentale d'une ontologie modulaire et son cycle de vie. Et la troisième section présente la mise en œuvre de cette méthode pour EDF et l'état actuel de l'ontologie de la $ME$.

## 2 Synthèse de l'état de l'art

Comme le montre la littérature récente sur les ontologies, la modularité est une des clés pour mieux développer, utiliser et ré-utiliser des ontologies. Les ontologies permettent de mieux partager l'information, ce partage se retrouve au cœur de nombreuses nouvelles technologies. Toutefois des besoins spécifiques à chaque application demeurent. Dans ces conditions, on compare la rentabilité d'une application spécifique par rapport à l'investissement dans une base d'informations communes. Force est de constater que le bilan est positif en faveur de l'investissement dans le partage d'informations, par exemple : internet, e-commerce, les moyens de paiement, etc. Dans ces domaines les acteurs se sont mis d'accord, entre autre, sur la façon de communiquer, pour leurs bénéfices mutuels. Aujourd'hui les industries sont de plus en plus connectées entre elles, on voit notamment l'émergence de projet comme les *smarts-grids*, *smart-cities*, etc. Il est donc naturel de voir des industriels s'intéresser de près à ces questions de partage d'informations pour développer leurs services.

Bien que les travaux de la communauté scientifique permettent maintenant de disposer de bases solides sur les ontologies, il reste néanmoins des verrous à lever pour permettre de plus nombreux déploiements d'ontologies industrielles. Les travaux cités sont produits par des experts sur les ontologies à destination d'un public familier avec les ontologies. D'ailleurs le terme *ontology engineer* est récurrent dans certains articles lorsqu'il est question de mettre en place des techniques pointues sur les ontologies. Bien que des ontologies puissent servir dans de nombreux domaines, les entreprises de ces domaines préfèrent centrer leurs compétences et leurs ressources sur leur cœur de métier. Ainsi les ontologies se développent peu sur des domaines où elles seraient utiles comme dans le cas de la $ME$. Pour résumer, il manque une méthodologie plus orientée vers les contraintes opérationnelles pouvant être mise en place avec un minimum de pré-requis sur les ontologies.

La méthode que nous proposons tient compte des problématiques opérationnelles en s'appuyant sur les travaux de modularisation déjà existants. De plus, plutôt que d'attendre un consensus général sur les modules, nous proposons une construction de l'ontologie incrémentale.

L'état de l'art sur les ontologies modulaires permet de s'appuyer sur des définitions robustes des modules. Toutefois il ne faut pas négliger le contexte industriel dans lequel nous nous trouvons. Les ingénieurs et chercheurs qui seront les futurs utilisateurs de l'ontologie sont des experts sur les différents aspects du domaine. C'est pourquoi nous avons choisi de baser la construction de l'ontologie sur leurs connaissances métiers.

# 3 Construction incrémentale d'une ontologie modulaire

Notre méthode consiste en un assemblage de *briques ontologiques*, une brique s'articule autour d'un concept clé et de l'éventuelle brique maîtresse.

Dans cette section nous définissons en premier lieu ce qu'est une brique ontologique, en précisant les différentes natures des briques nécessaires à notre méthode. Une fois les briques définies nous détaillons leurs constructions ainsi que les contraintes à respecter afin d'assurer la modularité. Enfin nous détaillons les techniques d'assemblage des briques et les possibilités offertes par notre méthode.

## 3.1 Définition d'une brique ontologique

Une brique ontologique est un ensemble de classes et de propriétés respectant des contraintes d'associations particulières. Nous détaillons ici les différents éléments formant une brique.

Une brique constitue une ontologie locale autour d'un seul concept, il s'agit de la **classe principale** de la brique. La classe principale constitue la raison d'être de la brique, l'unique objectif de la brique est de détailler ce concept.

En plus de la classe principale, la brique possède des **classes secondaires** dont le rôle est d'enrichir les propriétés de la classe principale. Les propriétés liant les classes secondaires à la classe principale font partie de la brique ontologique.

Pour un domaine en évolution il peut s'avérer nécessaire de spécialiser la classe principale. Pour cela une brique peut contenir des sous-briques, elles-mêmes étant des briques à part entière. Les classes principales des sous-briques **héritent** de la classe principale de la brique qui les contient.

La figure 3.1 illustre ces éléments.

On distingue deux types de briques : les briques du plus haut niveau, assimilables à une ontologie globale, et les briques spécialisées. Ces briques sont présentées dans les paragraphes

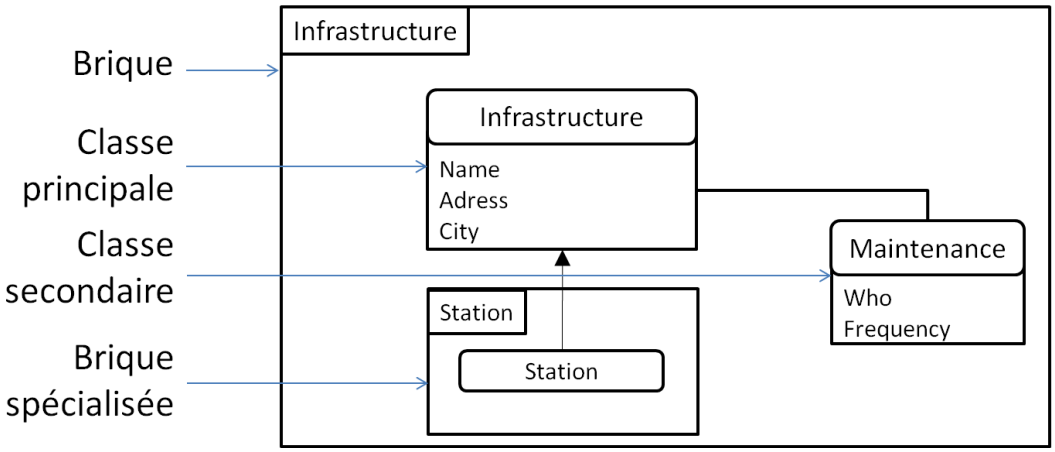


FIGURE 3.1 – Brique ontologique

ci-dessous.

Par rapport aux contraintes précisées dans l'état de l'art, cette construction des modules répond aux exigences d'EDF et des autres acteurs.

En effet, un industriel peut, grâce à ce type de brique, se contenter de la classe principale s'il n'a pas usage de cette brique. Ce premier point permet de mettre en œuvre l'ontologie globale sur les sous-domaines qui ne correspondent pas aux besoins de l'industriel. Celui-ci a la possibilité de détailler certaines briques pour être au plus près de ses besoins. L'utilisation de briques moins détaillées permettra à terme de garder les liens vers les modules des autres industriels. Ce point sera détaillé sur l'assemblage des briques.

Au sein même d'EDF cette démarche apporte une solution pratique, la R&D étant découpée en plusieurs départements, chacun ayant ses spécialités. Cet état de fait permet de tester directement la dualité ontologie globale-domaine entre les experts des différents départements.

Avec les classes secondaires chaque groupe d'experts peut développer le module qui lui correspond, toujours en suivant un cycle itératif pour chaque brique.

### 3.1.1 Briques du plus haut niveau

Les briques respectent une hiérarchie grâce au mécanisme d'héritage, les briques au sommet de ces hiérarchies sont les briques du plus haut niveau. Les concepts clés de ces briques n'héritent d'aucun autre et forment ensemble l'ontologie globale de la $ME$. Leur rôle est de structurer l'ensemble des briques. Chaque brique spécialisée hérite, directement ou indirectement, d'une de ces briques. Ce squelette est indispensable et nécessite d'être accepté par tous les acteurs. Le consensus est possible grâce à la nature du domaine : manipulé par des experts il est connu et exploité dans l'industrie. Cette exploitation globale est l'une des sources d'hétérogénéité mais elle permet également une division du domaine par corps de métier.

### 3.1.2 Briques spécialisées

Toute brique spécialisée hérite, soit directement soit à travers une succession d'héritage, d'une brique du plus haut niveau. Elle apporte une spécialisation d'un concept déjà existant pour approfondir une description, introduire des nuances ou répondre à un besoin expert spécifique.

## 3.2 Construction d'une brique ontologique

A partir des briques du plus haut niveau la méthode de construction adopte une approche descendante (*top-down*). La méthode tire partie de la présence des experts pour raffiner les briques du plus haut niveau en sous-domaines clairement définis. Le cycle de vie employé pour construire une brique est basé sur un cycle itératif classique que nous avons adapté.

La construction d'une brique est manuelle, elle démarre par l'expression d'un concept par les experts. Ce concept va être comparé aux éléments de l'ontologie afin de définir s'il doit devenir la classe principale d'une nouvelle brique ou s'il vient compléter la définition d'un concept déjà existant. Cette étape se fait avec les experts.

Si le concept est choisi pour compléter la définition d'un autre concept alors on l'ajoute à la brique de ce concept et on établit les propriétés entre les concepts. Sinon, on forme une nouvelle brique à partir du concept et on cherche la brique qu'elle spécialise. On est sûr de trouver une brique à spécialiser car on dispose d'une ontologie globale couvrant tout le domaine. Cette démarche est itérée pour chacun des concepts exprimés par les experts.

Comme indiqué dans la figure 3.1 une brique est centrée sur un concept : la classe principale. La phase de spécification de la classe principale va permettre de déterminer son périmètre et donc les propriétés de la brique. Ensuite l'analyse par un expert va permettre de déterminer si les propriétés sont des attributs de la classe principale ou si des classes secondaires sont nécessaires, c'est la phase de conceptualisation. Les constituants de la brique sont ensuite implémentés avec *PLIB*, le formalisme que nous avons retenu. Le cycle de vie est itéré jusqu'à ce que les experts soient satisfaits par la brique obtenue. La brique ainsi constituée va être directement exploitée par l'entrepôt de données (chapitre 4) et par les processus métiers (chapitre 5). A tout moment la brique peut être amenée à changer, toujours à travers un cycle de vie itératif, et nous présentons dans les chapitres suivants comment ces changements impactent le cycle de vie de l'entrepôt de données et celui des processus métiers.

Ce cycle n'inclut pas de phase de tests à proprement parler, il s'agit d'avantage d'une phase d'évaluation. Ainsi nous parlons de l'adaptation d'un cycle de vie itératif.

L'approche descendante adoptée donne lieu à la création de trois types de briques :
- *La brique maîtresse* : elle contient toutes les autres, sa forme est particulière car elle ne contient pas de brique principale et les sous briques sont reliées entre elles.
- *Les briques du plus haut niveau* représentent des concepts abstraits. Leurs concepts sont considérés stables et vont contenir d'autres briques. Les liens entre les champs d'expertise

doivent, autant que possible, être définis au niveau de ces briques.

– *Les briques de bas niveau* contiennent les concepts les plus spécifiques de l'ontologie. **Ce sont ces concepts qui permettront de palier aux problèmes d'hétérogénéité des sources de données.**

La création des briques s'est faite en discutant avec les experts et le laboratoire. Les experts apportent une vision claire et précise du domaine et de leurs besoins vis-à-vis des différentes briques. Les échanges avec le laboratoire ont permis de structurer l'ontologie avec la définition proposée dans la section précédente. L'itération de ces allers-retours a permis d'aboutir rapidement à des briques stables. Avec cette méthode un module peut être crée en quelques jours, au maximum quelques semaines, suivant la complexité de la brique. Avec la méthode détaillée dans le chapitre suivant (chapitre 4) les modules ont été opérationnels tout de suite et utilisés pour le stockage et l'analyse des données.

## 3.3 Assemblage des briques

Notre démarche est adaptée à l'efficacité opérationnelle recherchée par les experts. Chaque brique peut être conçue sans rechercher de consensus, ce qui permet à un expert de formaliser sa conception d'un champ d'expertise, et par la suite de bâtir un $SI$ dessus (voir chapitre 4). De cette façon, différentes briques peuvent être définies sur un même champ par différents experts. Cette dernière observation nous avait amenés à considérer une brique comme une ontologie locale.

L'ontologie globale est produite en assemblant manuellement les briques. Pour obtenir un résultat cohérent l'assemblage doit respecter plusieurs règles :

– Les liens *verticaux*, qui interviennent entre les classe principales de briques et les classes principales des sous-briques, sont assurés par des liens d'héritages. Cela permet d'assurer la spécialisation d'un sous-domaine vers les concepts atomiques.
– Les liens *horizontaux* requis entre certaines briques sont supportés par les classes principales des briques concernées. Cela revient donc à créer des propriétés supplémentaires.

Ces contraintes de relation entre les briques sont indispensables au bon assemblage des briques en vue de leur utilisation dans un $SI$. De plus, l'objectif étant de disposer d'une ontologie flexible afin que chaque expert puisse soit avoir ses propres briques soit en proposer des nouvelles il faut être capable d'ajouter ou de remplacer des briques pour faire évoluer l'ontologie. Cela devient possible en contraignant les relations entre les concepts selon les règles indiquées précédemment.

Faire porter les relations par les classes principales des briques permet notamment de modifier à loisir les composants descriptifs de la brique, cela sans porter préjudice à l'assemblage. La particularité des relations verticales (héritage) est de faciliter l'extension de l'ontologie par la spécialisation. Cela offre l'avantage supplémentaire de conserver les questions posées à l'ontologie et d'apporter plus de précisions aux réponses.

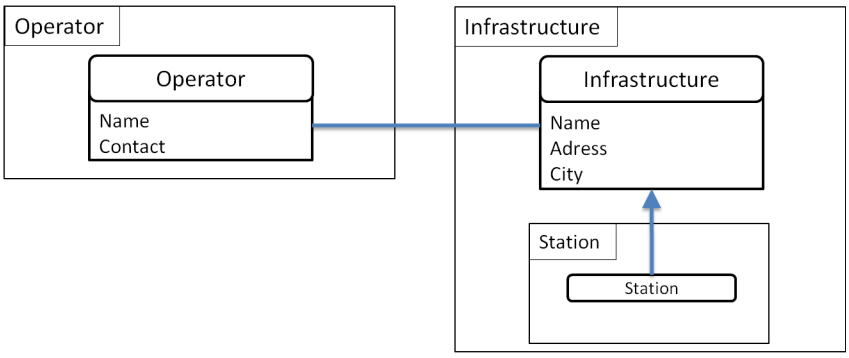


FIGURE 3.2 – Connexions des briques

Par conséquent on obtient la règle suivante : **pour remplacer une brique par une autre il faut assurer les liaisons avec les classes principales des autres modules**. Le contrôle de la consistance est quant à lui réalisé manuellement par les experts.

La section ci-dessous présente la mise en pratique de cette nouvelle approche pour créer l'ontologie de la *ME*.

Dans la pratique l'assemblage des briques a fait partie des allers-retours entre experts industriels et le laboratoire pour construire les briques. Les contraintes d'assemblages ont pu être éprouvées au sein de la R&D grâce aux interactions entre les départements. Ce test concluant a permis d'observer comment les experts de sous-domaines spécifiques ont pu exploiter les briques en dehors de leurs spécialités. Le choix de centrer les briques sur un concept clé a permis un meilleur partage de la structure de l'ontologie. Ensuite, dans un sous-domaine, les experts ont pu se focaliser sur leur cœur de métier. Cette capacité à rester vague mais compatible avec un sous-domaine distant tout en maximisant le temps passé sur les briques du sous-domaine a été particulièrement apprécié par les experts des sous-domaines et également par les différents acteurs concernés, comme les chefs de projets, les responsables commerciaux, les responsables des relations avec les partenaires ou les experts chargés d'assembler les briques des sous-domaines.

La section ci-dessous montre l'état actuel de l'ontologie telle qu'elle est partagée.

# 4 Ontologie de la ME

## 4.1 Éléments existants

Le domaine de la *ME* est étudié en France depuis plusieurs années par EDF ainsi que par différents industriels : les constructeurs de $\mathcal{VE}$, les constructeurs de bornes, les entreprises chargées des grands travaux de voiries, etc. Ces industriels ont formé un groupe de travail autour de la *ME* afin d'en définir les concepts clés. Cette approche est proche des méthodes classiques de

construction d'ontologies. Toutefois l'objectif n'est pas de baliser le domaine mais de définir les acteurs du domaine et certains équipements clés. Pour renforcer l'effort de standardisation ce groupe a rattaché les éléments sélectionnés aux normes existantes :

- La norme ISO/FDIS 15118 relative aux véhicules routiers et l'interface de communication entre le véhicule et le réseau électrique.
- La norme IEC 61850 est un standard pour la conception des automatismes des postes électriques.

D'autre part les expérimentations menées ont été l'occasion de mettre en place des bases de données pour collecter des données similaires. Ces tests ont permis d'acquérir de l'expérience sur le domaine et sur la façon de le décrire.

Dans cette situation nous disposons de deux types d'éléments qui facilitent la construction de l'ontologie selon la méthode proposée. D'un côté il existe des éléments qui sont déjà consensuels parmi les acteurs. De l'autre nous avons eu l'occasion de découvrir les spécificités des métiers des différents acteurs au cours des expérimentations.

Avec les éléments cités ci-dessus et avec la méthode proposée nous avons construit l'ontologie de la $ME$ avec des briques adaptées aux besoins d'EDF. Toutefois ces briques ont été prévues pour prendre en compte les besoins opérationnels rencontrés dans les premières expérimentations. Cela nous a amenés à créer des briques prenant en compte les besoins des partenaires.

Afin d'initialiser notre méthode nous nous sommes appuyés sur une division consensuelle du domaine pour choisir et définir les briques du plus haut niveau. Cette division est basée sur les métiers et les techniques de la $ME$. Ce point précis est rendu possible par la présence chez EDF de très nombreux métiers : depuis des experts sur les matériaux utilisés dans les batteries aux experts sur la mobilité des ménages et leur impact sur les réseaux électriques. Voici les principaux sous-domaines identifiés (voir figure 3.3) au terme de nos échanges avec différents corps de métiers (chez EDF comme chez ses partenaires) :

- *Stakeholder* : cette brique décrit les parties prenantes (*i.e.* : les acteurs) de la $ME$. Ce concept est abstrait et l'expérience a montré qu'il possède peu d'instance par rapport aux nombres d'acteurs. Toutefois les instances s'avèrent indispensables afin d'indiquer les destinataires d'une information, les responsables d'un service, etc.
- *Equipment* : cette brique regroupe tous les équipements, mobiles ou non, et les infrastructures. C'est le sous-domaine le mieux maîtrisé par EDF en raison de sa connaissance et de son expérience (publications, experts, brevets, laboratoires, etc.) en matière d'équipements électriques.
- *Data and Event* : cette brique permet de réunir toutes les descriptions des données générées et échangées ainsi que les évènements déclencheurs de ces échanges. C'est le sous-domaine qui permet d'impliquer les acteurs du domaine pour d'améliorer et d'étendre leurs services, les rendre les plus compatibles possibles, etc.

Les sections suivantes présentes l'état actuel de l'ontologie telle qu'elle est partagée entre les experts d'EDF et leurs partenaires sur la $ME$. Comme cela apparait plus haut les termes

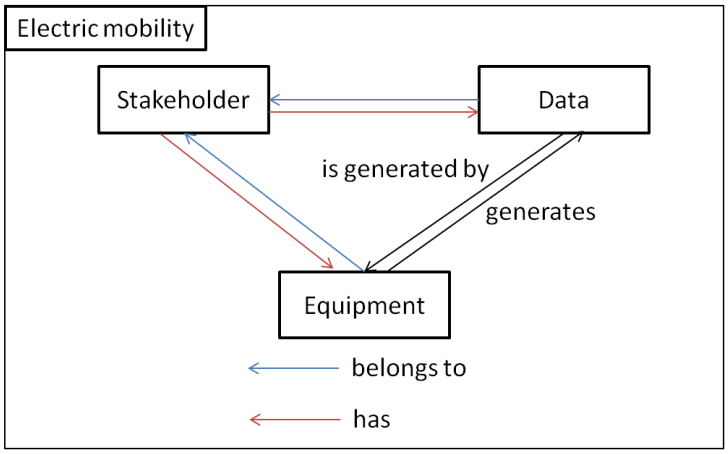


FIGURE 3.3 – Principales briques de l'ontologie de la *ME*

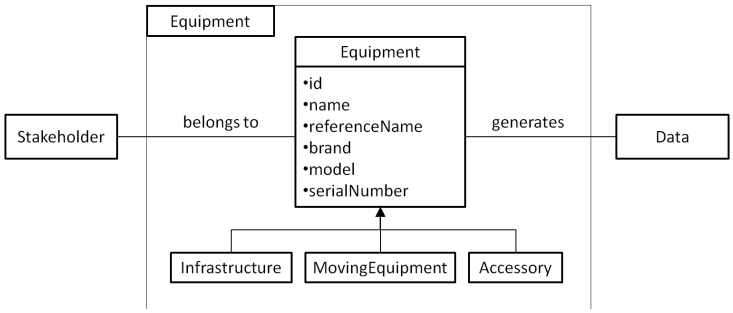


FIGURE 3.4 – Brique des équipements

employés sont des termes anglais ainsi les figures et les noms des briques sont présentés en anglais, les explications quant à elles utilisent les termes français. C'est un choix qui a été effectué en raison de l'aspect international de la *ME*.

## 4.2 Équipements

Un équipement est décrit par :
– un identifiant,
– un nom courant et un nom de référence,
– des informations commerciales : marque, modèle et numéro de série.

Un équipement peut être spécialisé en trois éléments qui forment chacun une sous brique :
– les infrastructures,
– les équipements mobiles,
– les accessoires.

La figure 3.4 représente la brique *Equipment*.

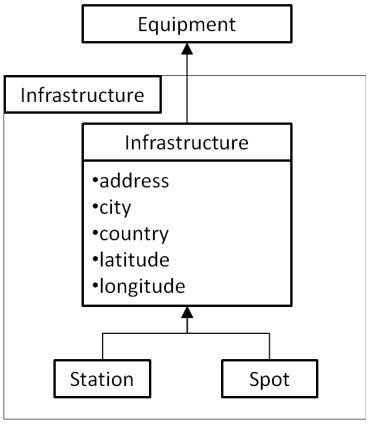


FIGURE 3.5 – Brique des infrastructures

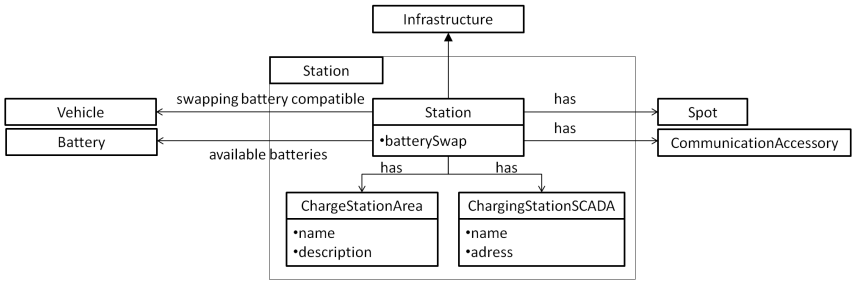


FIGURE 3.6 – Brique de l'élément station.

### 4.2.1 Infrastructures

Une infrastructure est un équipement caractérisé par une position physique (voir figure 3.5), c'est-à-dire :

- une adresse, une ville et un pays,
- des coordonnées GPS.

Une infrastructure possède deux spécialisations : station et borne.

**4.2.1.1 Station** Une station est une infrastructure qui rassemble des bornes de recharge (voir figure 3.6) superviser par un système de contrôle et d'acquisition de données (*SCADA*). La définition d'une station est ensuite étoffée par différentes caractéristiques :

- l'échange de batterie, les batteries disponibles le cas échéant et les véhicules compatibles,
- une capacité à communiquer en complément du *SCADA*.

**4.2.1.2 Borne** Une borne représente une infrastructure où un véhicule peut se charger, illustrée dans la figure 3.7. La borne est le **point d'interaction** entre l'utilisateur du $\mathcal{VE}$, le gestionnaire d'infrastructures, le fournisseur d'électricité et les *smart-grids*. Pour son fonctionnement, la borne se doit de pouvoir identifier l'utilisateur, les instruments de mesure et l'espace pour

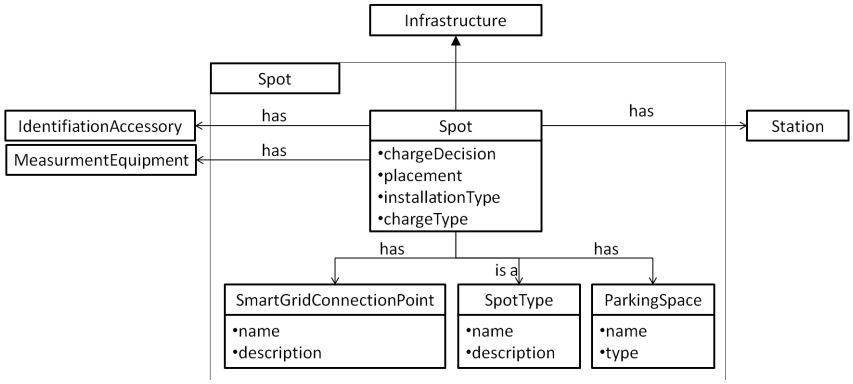


FIGURE 3.7 – Brique de l'élément borne

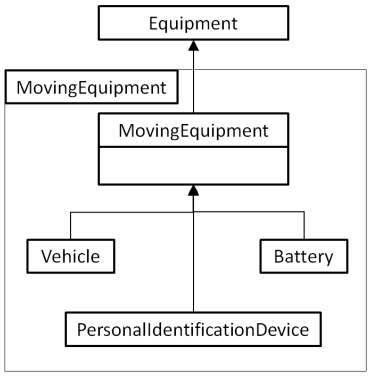


FIGURE 3.8 – Brique des équipements mobiles

accueillir le $\mathcal{VE}$.

### 4.2.2 Équipements mobiles

La brique sur les équipements mobiles existe pour conserver la hiérarchie des concepts d'une sous-brique à une autre. Cette brique est toutefois amenée à être enrichie par des propriétés de localisation. On retrouve trois sous-briques (voir figure 3.8) :

- Véhicule
- Moyen d'identification personnel
- Batterie (pour les cas d'échanges de batteries)

**4.2.2.1 Véhicule** Un véhicule est caractérisé par son usage et sa plaque d'immatriculation (voir figure 3.9). La nomenclature des usages a été établie par les analyses menées par les sociologues du projet $\mathcal{VE}$. On retrouve la liste suivante :

- $\mathcal{VE}$ de service, $\mathcal{VE}$ de service mono-utilisateur, $\mathcal{VE}$ de service multi-utilisateurs,
- $\mathcal{VE}$ de fonction,
- $\mathcal{VE}$ particulier.

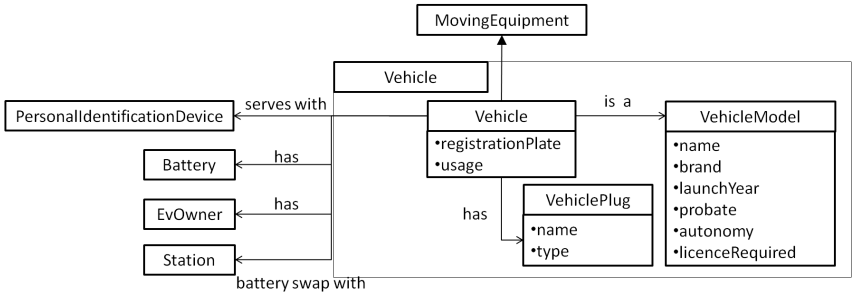


FIGURE 3.9 – Brique de l'élément véhicule

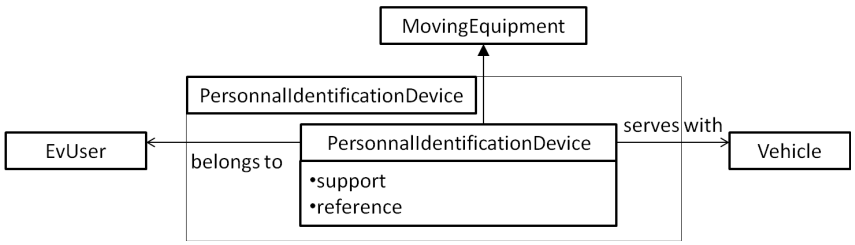


FIGURE 3.10 – Brique de l'élément moyen d'identification personnel

La description comprend également les badges des utilisateurs et le propriétaire. D'un point de vue matériel on retrouve dans la description du véhicule : sa batterie, les stations où l'échange de batteries est possible, la description du modèle de véhicule et les prises de charge dont est équipé le véhicule.

**4.2.2.2 Moyen d'identification personnel** Il est caractérisé par un support (cf figure 3.10) : dans la plupart des cas il s'agit d'un badge mais il peut également s'agir d'un code (support virtuel). En tant qu'équipement il dispose d'un identifiant (par héritage) toutefois on lui rajoute une référence qui représente ce qui est reconnu par les systèmes d'identification. On peut également le lier aux véhicules utilisés.

**4.2.2.3 Batterie** Une batterie est principalement décrite par ces caractéristiques intrinsèques (voir figure 3.11) : composition chimique, capacité (ou énergie stockable), poids et si elle peut être échangée dans une station. La description permet de savoir si la batterie est utilisée par un véhicule ou si elle est disponible dans une station.

### 4.2.3 Accessoires

Cette brique n'est pas au centre de l'attention d'EDF pour le moment, toutefois les éléments du *role model* y sont représentés (voir 3.12). La description d'un accessoire n'est pas faite par une relation de spécialisation (héritage) mais sous forme de propriétés (optionnelles).

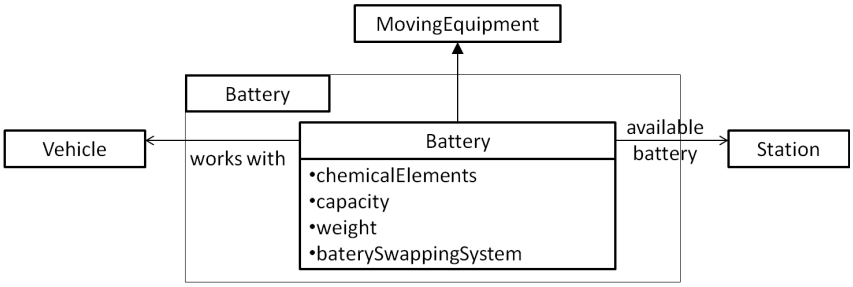


FIGURE 3.11 – Brique de l'élément batterie

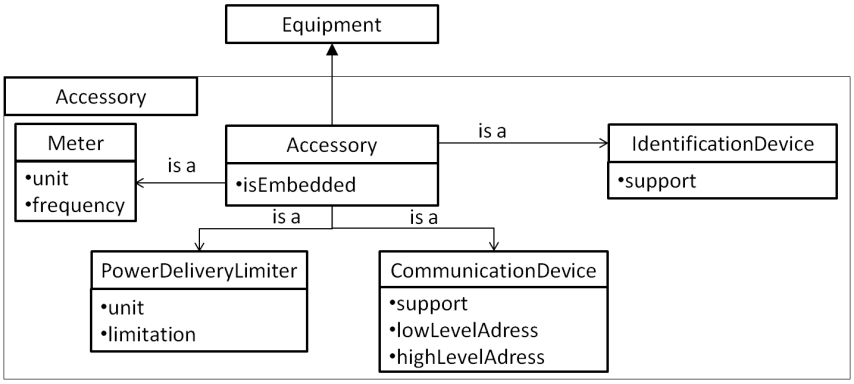


FIGURE 3.12 – Brique des accessoires

## 4.3 Données et évènements

Les données et les évènements constituent une brique primordiale (voir figure 3.13). Dans cette brique sont décrites les informations qui seront échangées entre plusieurs acteurs du domaine. C'est également parmi ces éléments que l'on va retrouver les concepts au cœur des analyses, comme la définition d'une charge.

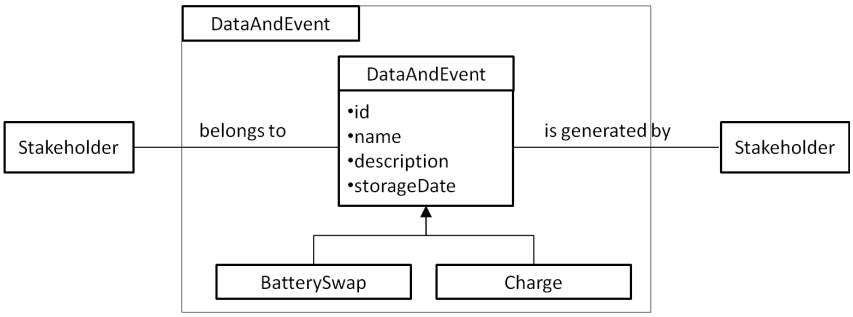


FIGURE 3.13 – Brique des données et des évènements

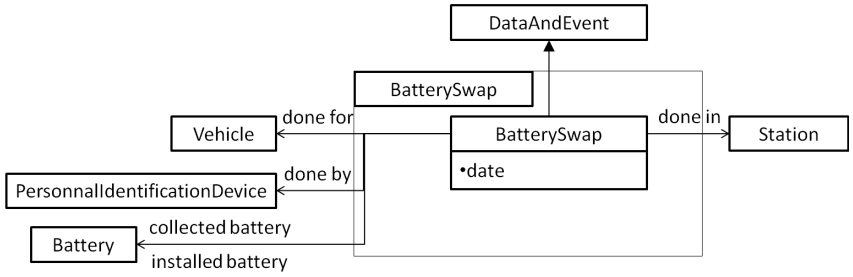


FIGURE 3.14 – Brique sur l'échange de batteries

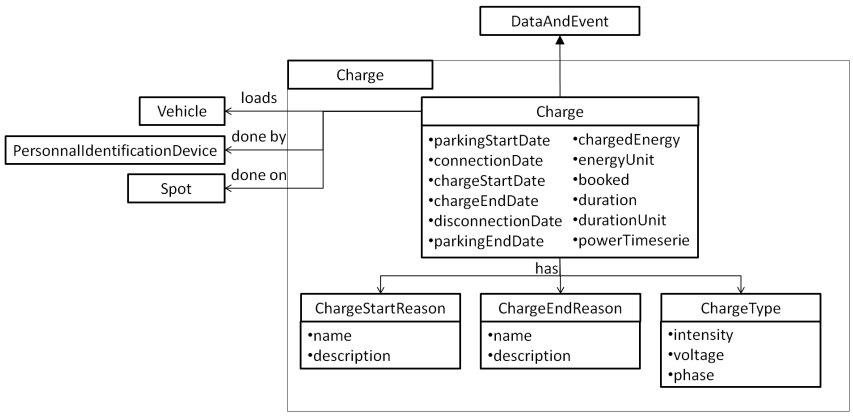


FIGURE 3.15 – Brique de la charge

### 4.3.1 Échange de batterie

Un échange de batterie désigne le remplacement de la batterie d'un véhicule par une batterie chargée dans une station [15] (voir figure 3.14). On retrouve dans sa description le $\mathcal{VE}$ concerné, son utilisateur (au travers de son moyen d'identification), les batteries échangées et la station dans laquelle l'échange est effectué.

### 4.3.2 Charge

Les charges représentent **la clé de voûte des analyses menées par EDF** car elles sont au centre de la $ME$ du point de vue du gestionnaire du réseau électrique. De fait, il s'agit d'un des concepts ontologique le plus détaillé (voir figure 3.15).

## 4.4 Parties prenantes

Les parties prenantes regroupent les acteurs de la $ME$ (voir figure 3.16), on y retrouve :
- les utilisateurs des $\mathcal{VE}$,
- les propriétaires des $\mathcal{VE}$,

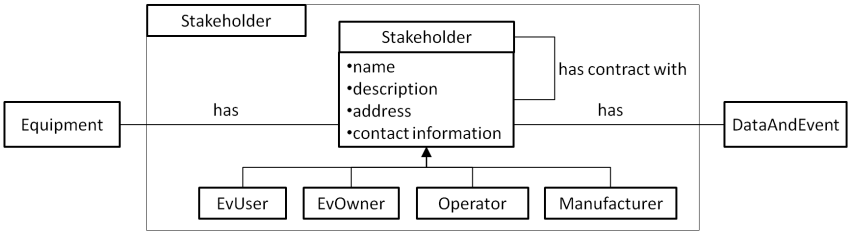


FIGURE 3.16 – Brique des parties prenantes à la *ME*

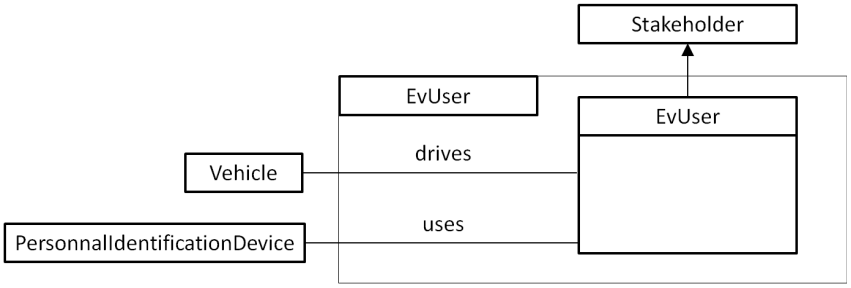


FIGURE 3.17 – Brique des utilisateurs des VE

– les opérateurs,
– les constructeurs.

### 4.4.1 Les utilisateurs

Ce sont les conducteurs des $\mathcal{VE}$ qu'ils en soient les propriétaires ou non (voir figure 3.17). Ils sont caractérisés par des moyens d'identification personnels et par les véhicules qu'ils utilisent.

### 4.4.2 Les propriétaires

Les propriétaires de $\mathcal{VE}$ (voir figure 3.18) possèdent des propriétés similaires à celles des utilisateurs à l'égard des $\mathcal{VE}$ et des moyens d'identification. Cette catégorie permet de définir les gestionnaires de flotte identifiés dans la nomenclature (issue d'un groupe de travail inter-industriels dont EDF fait partie) des acteurs du domaine.

### 4.4.3 Les opérateurs

Cette brique regroupe les acteurs offrant des services liés à la *ME* : contrat de recharge, gestion de parkings équipés de bornes, location, etc. A ce titre un opérateur peut avoir de nombreuses propriétés concernant les équipements (voir figure 3.19).

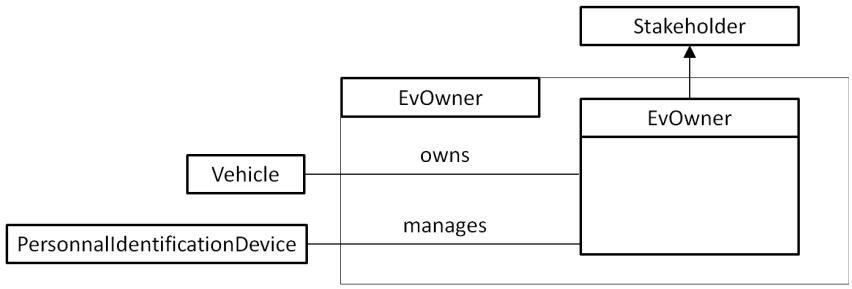


FIGURE 3.18 – Brique des propriétaires de VE

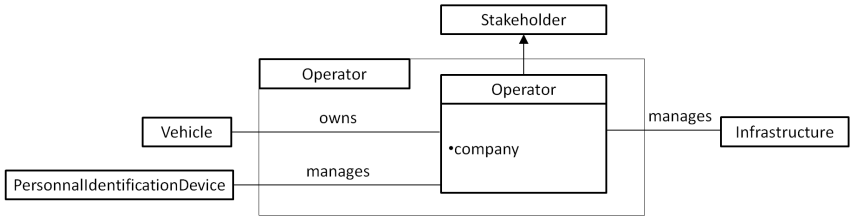


FIGURE 3.19 – Brique des opérateurs

### 4.4.4 Les constructeurs

Les constructeurs sont caractérisés par les équipements qu'ils produisent mais également par ceux dont ils assurent l'entretien (voir figure 3.20).

## 4.5 Synthèse

Cette version de l'ontologie possède peu de propriétés et certains concepts ne possèdent que des propriétés vis-à-vis d'autres concepts en plus de leur nom et de leur définition. Ceci est dû à plusieurs facteurs :

– L'objectif de nos travaux est de disposer d'un entrepôt de données. A ce titre certains concepts sont détaillés car ils vont permettre de former les tables des faits. Alors que

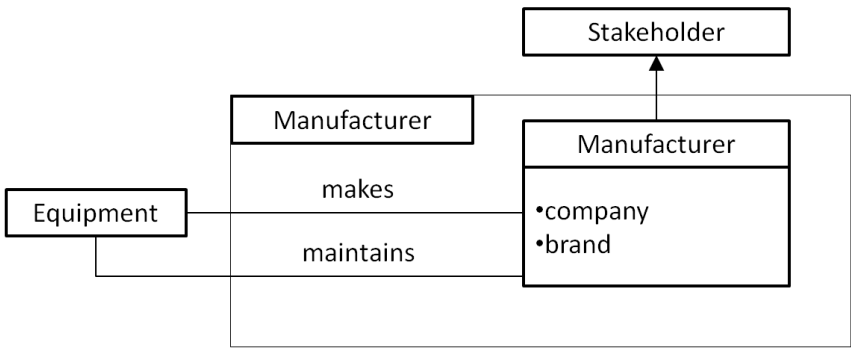


FIGURE 3.20 – Brique des constructeurs

d'autres concepts sont là pour détailler les dimensions, d'où le plus grand nombre de relations que de propriétés intrinsèques dans la majorité des concepts.
– Cette ontologie a été établie à partir des retours d'expériences acquis ces dernières années. Ainsi la situation initiale était basée sur des descriptions des données disponibles plutôt que par l'expression des besoins. Les éléments initiaux sont donc apparus *via* une approche *bottom up*.
– Enfin il s'agit des briques relatives à la vision de la R&D d'EDF et d'une partie de ses partenaires. D'autres visions sont en cours d'ajout.

# 5 Conclusion

Les ontologies possèdent un potentiel important pour des entreprises amenées à échanger de l'information. Elles peuvent aussi bien servir à bâtir des systèmes d'informations pour du stockage ou à utiliser des moteurs d'inférences. On peut également retrouver les ontologies pour faciliter les échanges d'informations ou gérer les problèmes d'hétérogénéité.

Malgré toutes les possibilités illustrées par une littérature scientifique riche, le recours aux ontologies n'est pas systématique dans le monde industriel. On en retrouve toutefois certains principes lors de l'élaboration de normes ou de standards. Il s'avère que créer une ontologie et la maintenir est une démarche coûteuse, comme nous l'avons vu dans l'état de l'art. Coûteuse en ressource d'experts : il faut disposer des experts nécessaires sur le domaine étudié et de spécialistes des ontologies pour tirer parti de l'expertise disponible. Couteuse en temps : la construction d'une ontologie requiert une analyse détaillée du domaine puis des discussions entre les experts pour aboutir à un consensus, cela peut prendre plusieurs années pour la construction d'une ontologie au sein d'une entreprise. Ainsi la création d'une ontologie, selon les méthodes classiques de construction, constitue un véritable investissement et engendre des coûts de maintenance.

Les méthodes de construction classiques s'appuient sur des hypothèses fortes relatives au domaine. Dans notre type de domaine - récent, non-mature et très partagé - ces hypothèses ne peuvent pas être vérifiées et pourtant une ontologie serait, justement, particulièrement utile. Plus récemment, le développement des ontologies modulaires permet d'alléger les contraintes afin de favoriser la ré-utilisation des modules dans différentes ontologies. Ces méthodes sont plus opérationnelles mais présentes néanmoins des inconvénients. Soit la définition des modules est contraignante et alors on se retrouve à terme avec une collection de modules plutôt qu'une ontologie, ce cas de figure entraîne des problèmes de relations entre les modules. Ou alors la définition des modules est plus libre et alors il devient difficile de les faire évoluer ou de les ré-utiliser.

La méthode de construction proposée **s'appuie sur des hypothèses réduites** afin de minimiser l'investissement à réaliser pour construire l'ontologie. La construction de l'ontologie avec les autres acteurs du domaine est réalisée de façon **incrémentale** en cherchant **localement**

**un consensus opérationnel**. De cette façon l'investissement est minimisé, facteur important, et l'ontologie en l'état est **exploitable immédiatement** avec les outils qui seront présentés dans les chapitres suivants.

L'ontologie proposée correspond aux besoins actuels d'EDF en termes de descriptions et d'analyses. Sa création a été faite itérativement en tenant compte des retours d'expériences des expérimentations d'EDF, des besoins des partenaires et des groupes de travail inter-industriels sur la $ME$. Cette version est amenée à évoluer en suivant la méthode proposée afin de refléter la vision de la $ME$ d'EDF et de ses partenaires.

En tant que telle l'ontologie sert de référence pour EDF en interne comme en externe, par exemple : les derniers formats d'échanges mis en place (projet CROME, présenté dans le chapitre 6) sont basés dessus. Au delà de proposer une référence et un outil de communication l'ontologie a surtout servi à améliorer le système de stockage de la mine de données disponibles. Le chapitre suivant (chapitre 4) présente comment nous avons adopté les méthodes de création d'entrepôts de données à base ontologique avec notre *design* particulier d'ontologie.

Chapitre 4

# Entrepôt de données à base ontologique

## Sommaire

# 1 Introduction

Pour rappel, l'objectif final d'EDF est de pouvoir générer des rapports sur les comportements des utilisateurs des $\mathcal{VE}$, à travers des analyses statistiques et des fouilles de données. Les analyses à mener portent sur des axes connus par les experts du domaine afin d'aider à la prise de décisions, comme par exemple le développement de nouvelles infrastructures là où la consommation est importante ou le choix des modèles de facturation. C'est pourquoi nous avions fait le choix d'un entrepôt de données ($ED$). Ce choix était également motivé par la capacité d'un $ED$ à supporter des volumes de données importants.

Pour pouvoir exploiter efficacement un $ED$ il faut pouvoir disposer de **données homogènes et de qualités**. Maintenant que nous disposons d'une ontologie nous pouvons répondre aux problèmes rédhibitoires d'hétérogénéité (voir l'état de l'art, chapitre 2). D'autre part, nous avons vu dans l'état de l'art que la création d'une ontologie, et donc de la formalisation du domaine, permet également de dégager des éléments utiles au cycle de vie de conception d'un $ED$. Nous disposons donc à présents des conditions requises en terme de qualité de données et de définition du domaine pour construire un $ED$. Il faut cependant adapter les méthodes exploitant les ontologies pour concevoir les $ED$ à notre forme d'ontologie et à son caractère flexible.

L'ontologie a servi à homogénéiser les données et à améliorer leur qualité grâce à une description du domaine, et ce faisant nous disposons d'une description commune des sources de données. Grâce à la notion de briques la description du domaine contient des hiérarchies ainsi que des liens entre les concepts. Ainsi, une grande partie des éléments nécessaires à la conception d'un $ED$ sont réunis, on connait : **les dimensions**, et **leurs descriptions**, ainsi que **les tables des faits**.

La construction d'un **entrepôt de données à base ontologique** ($EDBO$) est donc motivée par le gain en qualité des données et par une conception facilitée. Et ce n'est pas le seul avantage, dans le chapitre précédent nous précisions que les méthodes de modularisation des ontologies sont principalement destinées à des experts en ontologie. Ce constat nous avait influencé dans la création de notre méthode de construction d'une ontologie. Nous travaillons avec les experts du domaine, dans notre cas la mobilité électrique ($ME$), et le constat fait précédemment est toujours valable. En choisissant de créer un $EDBO$ on se base sur l'ontologie créée par les experts, et **l'ontologie devient alors l'interface pour réaliser le cycle de vie d'un $ED$ et l'utiliser** (voir figure 4.1).

Ce chapitre se décompose en trois sections pour présenter la méthode adoptée, depuis la conception de l'$ED$ à partir de l'ontologie jusqu'à son implémentation. La section 2 s'attache à décrire l'adaptation du cycle de vie de construction d'un $ED$ à partir de celui de l'ontologie telle que nous l'avons créée dans le chapitre précédent. La section 3 se focalise sur le problème du chargement de données dans le cas d'une évolution incrémentale de l'$ED$ du fait de la méthode de création de l'ontologie. Enfin la section 4 présente l'implémentation de ces concepts dans un système de gestion de base de données ($SGBD$) capable d'exploiter une ontologie. La section

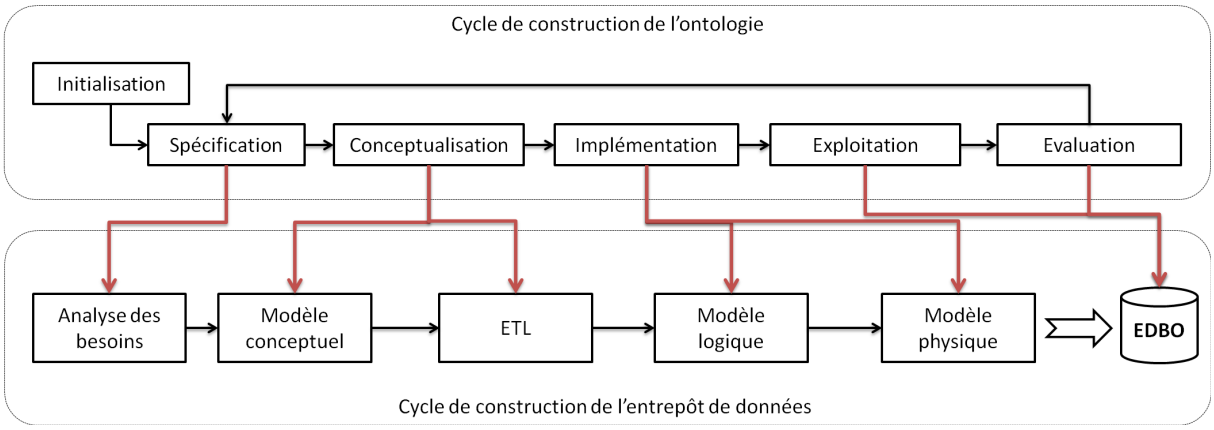


FIGURE 4.1 – Interaction entre le cycle de vie de création d'une ontologie et de conception d'un entrepôt de données

4 décrit également l'*EDBO* obtenu par la méthode proposée dans notre cas d'étude.

# 2 De l'ontologie modulaire à l'*EDBO*

L'état de l'art, dans le chapitre 2, détaillait les étapes de la conception d'un *ED* à travers la définition des besoins et des différents modèles. Il présentait également comment exploiter des ontologies dans la création de bases de données et d'*ED*. Il se terminait en constant que ces travaux supposent l'existence d'une ontologie.

A travers le chapitre 3 nous avons vu comment créer une ontologie dans notre contexte industriel, avec des contraintes opérationnelles génériques. L'ontologie dont nous disposons à présent a été conçue à partir de la méthode détaillée dans le chapitre 3, et elle est amenée à évoluer avec le domaine. C'est pourquoi il est nécessaire d'adapter la méthode de conception et d'exploitation d'un entrepôt de données à base ontologique (*EDBO*) à une ontologie modulaire et à son évolution incrémentale.

De plus la démarche n'est pas destinée à un public d'informaticiens. Cet état de fait avait déjà orienté les choix lors de la conception de la méthode de création d'une ontologie (avec la définition des briques par les experts du domaine). Cet élément doit donc être à nouveau intégré dans la suite des démarches pour constituer un véritable prolongement de la méthode initiée.

## 2.1 Intérêt de la modularité dans la conception d'un *EDBO*

Les *ED* servent de supports à la prise de décisions. Or dans un domaine où les sources de données sont de plus en plus nombreuses, et où les analyses gagnent en précision, davantage d'informations et de concepts sont requis. Dans ces conditions la tendance s'oriente vers une

augmentation du nombre de concepts et de propriétés dans l'ontologie pour continuer à fournir aux décideurs le support nécessaire. Comme on dispose d'une ontologie modulaire ces changements, ou ajouts, sont **réalisables rapidement et à moindres coûts**. Il faut donc veiller à pouvoir projeter ces changements sur l'*EDBO* pour le garder à jour, que ce soit recharger des données ou en charger des nouvelles.

La structure en brique offre également la possibilité de remplacer une brique (nouvellement créée) par une autre, à la condition de respecter certaines contraintes sur les relations (indispensables au maintien de la cohérence). Il est donc aisé pour un expert d'utiliser une brique plus spécialisée vis-à-vis de ses besoins pour un usage précis et de bénéficier du reste de l'ontologie. La démarche inverse est également utile pour réaliser un système d'information ($SI$) plus généraliste, utilisant des briques moins spécifiques.

Ceci permet une hyper-spécialisation locale de l'ontologie tout en maintenant des capacités d'échanges. Il devient alors intéressant de formaliser l'extension d'une ontologie en briques en *ED*, afin de créer un *EDBO* sur mesure pour une activité, un expert ou un groupe d'experts. Dans la pratique le choix a été fait de construire et de maintenir un seul *ED* au sein d'EDF et d'utiliser les briques les plus spécifiques.

## 2.2 Du cycle de vie de l'ontologie au cycle de vie de l'entrepôt de données

### 2.2.1 Définition des besoins

La première étape de création d'un *ED* consiste, comme décrit dans le chapitre 2, à définir les besoins, c'est-à-dire à quelles questions l'*ED* devra-t-il répondre. Ainsi les besoins couvrent un large spectre de questions, à commencer par ce que l'on veut obtenir comme résultats jusqu'aux attributs des concepts à analyser, de façon à être exhaustif. L'ontologie permet justement de décrire le domaine de façon exhaustive et les besoins des utilisateurs portent sur l'analyse du domaine. **Ainsi la démarche consistant à formaliser le domaine permet dans le même mouvement de formaliser les demandes des experts.** Les besoins vont s'exprimer en terme d'analyse de concepts en fonction d'autres concepts. L'ontologie étant conçue par les experts, pour leurs besoins spécifiques grâce aux briques, tous les concepts et les relations entre eux sont présents. Ainsi les questions auxquelles doit répondre l'*ED* sont les questions que les experts se sont posées au moment de créer les briques ontologiques. Ceci afin que les données décrites répondent à leurs besoins. La question de la définition des besoins pour l'entrepôt est donc implicitement embarquée dans la création de l'ontologie.

Par exemple pour définir dans l'ontologie ce qu'est une *charge* les différents experts interrogés ont mentionné leurs besoins : «il nous faut telle et telle information». Nous avons donc repris ces réflexions de l'étape de *spécification* du cycle de vie de l'ontologie pour l'étape *définition des besoins* du cycle de vie de l'*ED*.

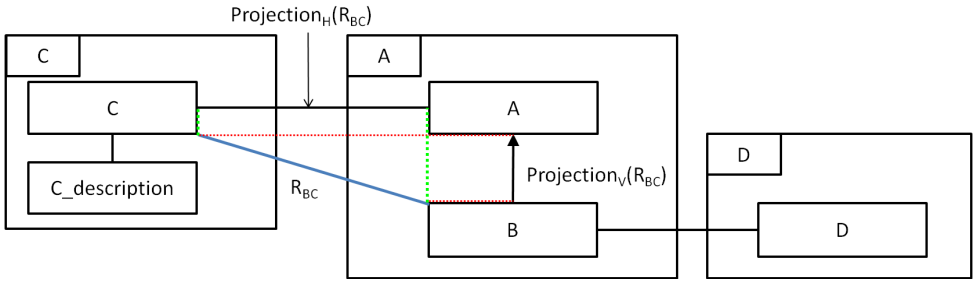


FIGURE 4.2 – Constructions des relations entre les concepts à partir des relations entre les briques

### 2.2.2 Appui à la création des différents modèles

**2.2.2.1 Le modèle conceptuel** Le modèle conceptuel, indépendant de l'implémentation de l'*ED*, va définir comment les données vont être stockées et comment seront formulées les requêtes.

La création du modèle conceptuel commence par la définition des entités : leur **nom** et leurs **attributs**. Or ces éléments sont présents dans l'ontologie, il s'agit des concepts ontologiques et de leurs propriétés. Ils ont été spécifiés dans l'étape du même nom du cycle de vie de l'ontologie (ou du module concerné). A partir de l'ontologie il suffit de sélectionner les concepts à faire figurer dans le modèle conceptuel pour en démarrer la conception. Cette première opération permet de définir le périmètre de l'*EDBO*.

Ensuite, dans l'ontologie les relations entre les concepts sont décrits de manière exhaustive : nom, cardinalité, concept de départ et d'arrivée. Plus précisément ce sont les relations horizontales qui vont être exploitées : ce sont ces relations qui relient les concepts d'une hiérarchie, ou d'une imbrication, à une autre. Les relations d'héritage quant à elles permettent d'exploiter les relations horizontales héritées. Ces relations permettent de relier les entités préalablement sélectionnées.

Les différentes relations sont illustrées sur la figure 4.2 : la relation des concepts *B* et *C* est obtenue grâce à l'héritage par *B* de la relation entre *A* et *C*.

Toutefois cela peut imposer l'utilisation d'entités uniquement pour assurer les relations entre deux entités. Pour reprendre l'exemple de la figure 4.2 si l'on souhaite relier *C* et *D* il faut nécessairement ajouter *B* au modèle conceptuel. Ce cas de figure est relativement rare dans la mesure où les experts connaissent les données disponibles et les liens nécessaires. Cette même connaissance guide, à la base, la création de l'ontologie.

La figure 4.3 illustre le cas général décrit ci-dessus. Pour relier une *charge* à l'*opérateur* d'une *borne* particulière les contraintes que nous imposons aux relations ne permettent pas de relier ces concepts directement. Il faut nécessairement passer par le concept de *borne*.

Concernant les classes de description figurant dans les briques pour enrichir les classes principales, elles peuvent figurer dans le schéma conceptuel de deux façons. Dans la première

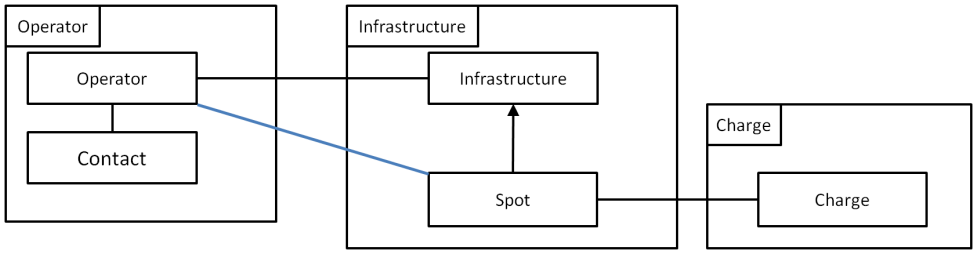


FIGURE 4.3 – Exemple de construction des relations entre les briques

méthode, les attributs de ces classes peuvent être intégrées à la classe principale, cette approche relève de la création du modèle physique. La seconde méthode consiste à faire figurer ces classes dans le modèle conceptuel. Dans le cas d'un *ED* les utilisateurs doivent décider si une telle classe fait partie d'un axe d'analyse. Par exemple, dans le cas d'un véhicule est-ce que faire figurer le modèle du véhicule dans un axe d'analyse est utile ou non ? Si c'est le cas la classe peut être ajoutée à la dimension où figure la classe principale en conservant une unique relation avec la classe principale. On se retrouve alors avec un **schéma conceptuel en flocon**. Pour illustrer ce choix d'après la figure 4.2, si *C* fait partie d'une dimension avec *B* et que l'on souhaite disposer de $C - description$ indépendamment de *C* pour les analyses alors on obtient la branche suivante : *B* - *C* - $C - description$. Mais il n'est pas possible d'avoir une branche *B* - $C - description$, comme précisé dans la section précédente.

A nouveau pour illustrer, la figure 4.3 donne un exemple de ce cas général. Pour relier une *charge* à la *description de l'opérateur* afin de facturer directement l'utilisateur il faut avoir le concept *opérateur* pour disposer du lien.

**2.2.2.2 Le modèle logique et le modèle physique** L'ontologie guide également la création du modèle logique. Tel que nous avons bâti l'ontologie tous les concepts sont des spécialisations d'une poignée de concepts de base qui sont dans les briques du plus haut niveau. Les concepts de ces briques possèdent peu d'attributs mais ces derniers concernent l'identification du concept (référence du produit, identifiant, nom, etc.) et sa description. Ces attributs sont ensuite hérités par tous les concepts du domaine, ainsi on dispose automatiquement des clés primaires pour chacun des concepts de l'ontologie et cela permet, également, de définir les clés étrangères.

Dans le cas des classes de description utilisées dans les dimensions de l'*ED* il faut veiller à définir les clés primaires, souvent ces classes possèdent un attribut adéquat, pour reprendre l'exemple cité un peu plus tôt : le modèle d'un véhicule constitue en soi une clé primaire éligible.

L'ontologie décrit précisément tous les types de données (chaîne de caractères, entiers, etc.) indépendamment de la méthode de construction. Toutes les informations nécessaires à la création du modèle physique sont donc disponibles au sein de l'ontologie.

### 2.2.3 Intégration des données

Face à un modèle conceptuel amené à évoluer et à une augmentation du nombre de sources avec des modèles de données potentiellement tous différents, il est prioritaire de s'intéresser à la question du chargement de données. Les problématiques liées à l'$ETL$ et la solution que nous proposons sont décrites dans la section suivante, voici tout d'abord une première synthèse.

## 2.3 Synthèse

On observe ici que la structure en brique adoptée lors de la construction de l'ontologie facilite deux points précis de la conception d'un $EDBO$ :

- D'abord la définition des besoins car le domaine est divisé par les experts en fonction de leurs connaissances et des analyses qu'ils souhaitent effectuer sur les données. Cette étape fait partie des étapes clés lors de la conception d'un $SI$, que ce soit en travaillant avec un sous-traitant ou tout simplement lors des échanges entre les différents experts.
- Ensuite la construction du modèle conceptuel *via* les contraintes imposées par les relations entre les briques. Cela offre un garde-fou utile et permet de gagner du temps.

La conception des modèles logiques et physiques découle des propriétés des ontologies, hormis l'héritage généralisé des clés primaires de notre méthode. Ainsi on est maintenant capable de créer des $ED$ hyper-spécialisés à moindre coût, ce qui est un aspect recherché par les experts de différents domaines dans le projet $\mathcal{VE}$.

# 3 ETL Sémantique

Comme nous l'avons indiqué dans le chapitre 2, le processus $ETL$ est une phase critique dans le cycle de vie de conception d'un $ED$ [42] et coûteuse en temps [115, 137]. La qualité des analyses faites à partir d'un $ED$ dépend fortement de cette phase. Un nombre important de travaux sur les $ETL$ existe, accompagné d'outils commerciaux comme *Oracle Data Integrator*, *Talend*, *Microsoft SQL Server Integration Services* et académiques [138]. La maturité de ces travaux a permis de définir une algèbre regroupant l'ensemble des opérations nécessaires pour le développement du processus $ETL$ [121]. Elle regroupe douze opérateurs présentés ci-dessous. Afin de faciliter leur description, nous supposons l'existence d'une source de données $S$ contenant des tables relationnelles.

- *Retrieve(*$S$*)* : récupère les enregistrements d'une source $S$ ;
- *Extract(*$Ins, T$*)* : extrait des enregistrements $Ins$ d'une table $T$ d'une source donnée ;
- *Merge(*$Ins_i, Ins_j$*)* : fusionne deux ensembles d'enregistrements ;
- *Filter(*$Ins, T$*)* : filtre les enregistrements qui sont conformes au schéma d'une table $T$ ;
- *Convert(*$T_i, T_j$*)* : convertit les enregistrements issus de la table $T_i$ en enregistrements conformes à la table $T_j$ ;

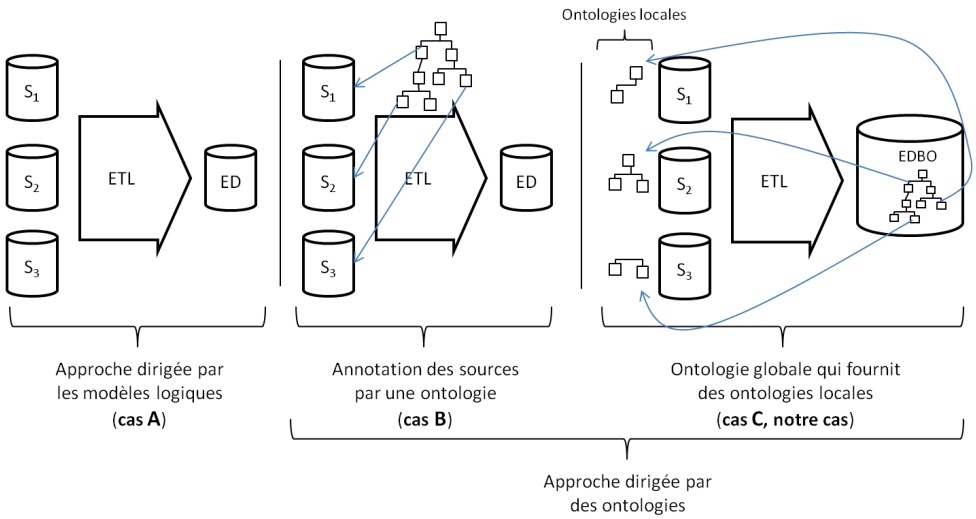


FIGURE 4.4 – Approches $ETL$

- *Aggregate(Ins, f, A1, ..., An)* : agrège les enregistrements sur les attributs $A_1, ..., A_n$ en y appliquant la fonction $f$ ;
- *MinCard(Ins, p, min)* : filtre les enregistrements ayant une cardinalité inférieure à un minimum sur la propriété $p$ ;
- *MaxCard(Ins, p, max)* : filtre les enregistrements ayant une cardinalité supérieure à un maximum sur la propriété $p$ ;
- *Union(Ti, Tj)* : établit l'union des enregistrements issus de deux tables $T_i$ et $T_j$ ;
- *DD(Ins)* : détecte et supprime les doublons présents dans un ensemble d'enregistrements ;
- *Join(Ti, Tj)* : réalise la jointure entre deux tables $T_i$ et $T_j$ ;
- *Store(Ins, Cible)* : stocke les enregistrements dans la cible.

En examinant la littérature, nous avons identifié deux principales catégories de travaux $ETL$ (figure 4.4) : (i) des *approches dirigées par les modèles logiques des sources* et (ii) des *approches dirigées par des ontologies*. Cette classification est basée sur le niveau de modélisation des paramètres des opérateurs de l'algèbre $ETL$.

**Approches dirigées par les modèles logiques** Dans la première génération des travaux sur les $ETL$, l'ensemble des opérations est défini sur les modèles logiques des sources de données [139]. L'inconvénient majeur de ces approches est qu'elles exigent la connaissance préalable des implémentations physiques de chaque source de données. En conséquence, une définition de $N$ instances d'opérateurs pour l'ensemble des sources est nécessaire (où $N$ représente le nombre de sources). L'hétérogénéité syntaxique et sémantique qui peut exister entre les sources est traitée manuellement, en exploitant l'expertise des concepteurs et la connaissance des sources de données.

Formellement la signature du processus $ETL$ dans ces approches a la forme suivante : $ETL : \{(S_1, SL_1), ..., (S_n, SL_n)\} \rightarrow ED$ où $S_i$ et $SL_i$ représentent respectivement la source de

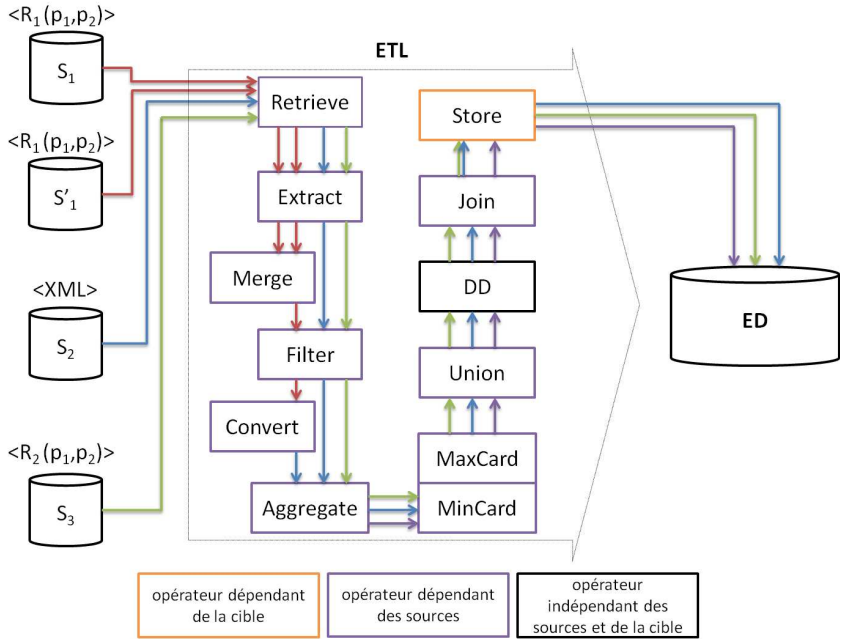


FIGURE 4.5 – Approche classique du processus $ETL$

données $i$ et son modèle logique qui peut être relationnel, $XML$, fichier, etc. La figure 4.5 illustre ces approches.

**Approches dirigées par des ontologies** Avec l'émergence des ontologies, la communauté de recherche a fait appel à cette technologie pour résoudre les conflits qui peuvent exister entre les sources participant aux processus de construction d'$ED$. Dans ce cadre, les ontologies ont été utilisées pour annoter l'ensemble des sources [61]. Dans ces travaux, une seule ontologie est utilisée par le processus $ETL$ (voir figure 4.4, cas B).

Avec le développement des sources de données sémantiques (où chaque source contient une ontologie locale qui référence une ou plusieurs ontologies de domaine) et vu la similarité entre les ontologies et les modèles conceptuels, nous proposons de redéfinir les opérateurs $ETL$ traditionnels au niveau sémantique. La présence de l'ontologie permet à la fois de rendre ces opérateurs génériques (indépendants des sources) et de résoudre les différents conflits. L'ontologie peut par exemple expliciter les unités de mesure utilisées dans les sources évitant ainsi les conflits de mesure. Cette situation a déjà été vécue lors de la conception des systèmes d'intégration ontologiques, qui supposent l'existence des ontologies locales au niveau des sources qui référencent une ou plusieurs ontologies partagées [87].

Un autre élément que nous souhaitons souligner concerne l'hypothèse faite sur la construction de l'ontologie. *Est-ce que l'ontologie a été construite pendant l'élaboration du projet d'entreposage ou est-elle supposée existante avant le projet ?*
Dans la majorité des travaux existants, cette question ne se pose pas, car elle suppose l'existence d'une ontologie. Dans notre travail, l'ontologie est créée avec les partenaires pendant l'élabora-

tion du projet d'entreposage. En conséquence, nous considérons que l'ensemble des partenaires la connaît parfaitement et que chacun a son ontologie locale (voir figure 4.4, cas C). Cette situation rend facile la génération des correspondances entre les ontologies locales et l'ontologie partagée contrairement aux approches traditionnelles, où des techniques de correspondance d'ontologies sont nécessaires [34].

La feuille de route que nous avons suivie pour développer notre $ETL$ sémantique comporte deux directions principales : (1) la re-définition des opérateurs au niveau ontologique et (2) leur implémentation.

## 3.1 Définition des opérateurs au niveau ontologique

Dans cette section, nous redéfinissons les opérateurs de base du processus $ETL$ au niveau ontologique. Rappelons que dans notre cas chaque source de données $S_i$ contient sa propre ontologie locale $OL_i$ et que l'$EDBO$ contient l'ontologie globale ($OG$). Dans ce cas, le processus $ETL$ se définit comme suit :

$\{(S_1, OL_1), ..., (S_n, OL_n), OG\} \rightarrow EDBO$

Lors de la redéfinition de l'ensemble des douze opérateurs, nous avons remarqué que certains peuvent être fusionnés, comme par exemple les deux opérateurs *Retrieve* et *Extract*. L'opérateur *Retrieve* permet de récupérer les enregistrements contenus dans une source. Les enregistrements ainsi récupérés sont fournis à l'opérateur *Extract* qui va sélectionner ceux qui sont conformes à une référence (comme une table pour les bases de données relationnelles). Dans notre cas, les données contenues dans une source sont décrites par une ontologie. Lors de leur récupération, on connait parfaitement leur origine. Devant cette situation, nous avons défini un autre opérateur fusionnant ces deux opérateurs, appelé *Retrieve*$(S_i, OL_i)$. Voici la définition ontologique des autres opérateurs :

- *Merge*$(Ins_i, OL_i, Ins_j, OL_j)$ : cet opérateur permet la fusion des ensembles d'instances $Ins_i$ et $Ins_j$ conformes respectivement aux ontologies locales $OL_i$ et $OL_j$.
- *Filter*$(Ins, C, C')$ : cet opérateur permet la sélection des données qui vérifie un certain nombre de contraintes. L'objectif est de charger uniquement les données requises dans l'$EDBO$. Au niveau ontologique, l'opérateur *Filter* permet la sélection des instances associées à la classe $C'$ de la source $S$, autorisant uniquement les instances correspondant à la contrainte spécifiée par la classe $C$ de l'ontologie de l'$ED$. Les contraintes sont définies au niveau de l'ontologie de l'entrepôt sous la forme d'axiomes, par exemple par des axiomes de cardinalités min et max ou par des énumérations.
- *Convert*$(instances, C, C')$ : la conversion est le mécanisme utilisé pour modifier les types et formats des données, plus précisément le format des attributs associés aux entités. Au niveau ontologique, la conversion correspond à la modification du format des instances associées à la classe $C'$ d'une source de données vers le format spécifié par la classe $C$ de l'ontologie de l'$ED$. La conversion s'effectue sur la base des propriétés (*DataType* pour

une ontologie *OWL*).

- *Aggregate(Ins, F, C, C′)* : l'agrégation est le mécanisme utilisé pour regrouper des données à des fins d'analyse. Ce mécanisme agrège les données selon certaines fonctions d'agrégation comme : *SUM, AVG, MAX, MIN* ou *COUNT*. Au niveau ontologique l'agrégation consiste à appliquer la fonction $F$ sur les instances associées à la classes $C'$ d'une source, et d'associer le résultat à la classe $C$ de l'ontologie de l'$ED$;
- *MinCard(Ins, p, min)* : cet opérateur permet d'appliquer une contrainte sur les instances de cardinalité minimale. Au niveau ontologique, la contrainte est appliquée sur la propriété $p$ où elle est définie par un axiome *min-cardinality*.
- *MaxCard(Ins, p, max)* : cet opérateur permet d'appliquer une contrainte sur les instances de cardinalité maximale. Au niveau ontologique, la contrainte est appliquée sur la propriété $p$ de l'ontologie de l'entrepôt où elle est définie par un axiome *max-cardinality*.
- *Union(C, C′)* : cet opérateur est utilisé pour fusionner deux entités appartenant à différentes sources de données. Au niveau ontologique, ce mécanisme correspond à l'union des instances associées aux classes $C$ et $C'$ appartenant respectivement aux sources $S$ et $S'$. Ces classes doivent être identiques ou avoir la même super classe.
- *DD(Ins)* : cet opérateur ne change pas en passant au niveau ontologique, il permet de détecter des doublons dans les instances et de les supprimer.
- *Join(C, C′)* : cet opérateur permet de joindre les instances des classes $C$ et $C'$ selon la propriété ontologique qui les relie.
- *Store(Ins, S, C)* : Au niveau ontologique, ce mécanisme correspond à l'affectation des instances *Ins*, récupérées à partir un ensemble de sources $S$, à la classe $C$ de l'ontologie de l'$ED$.

La présence des ontologies locales au niveau des sources et de l'ontologie globale au niveau de l'$EDBO$, nous a conduit à créer deux nouveaux opérateurs pour les manipuler :

1. **Retrieve_ontology(*EDBO*)** : cet opérateur permet de récupérer l'$OG$ embarquée dans l'$EDBO$ ;
2. **Retrive_source_local_ontology(*S*)** : cet opérateur permet de récupérer l'ontologie locale de chaque source.

Nous disposons à présent de tous les éléments nécessaires pour implémenter notre processus $ETL$.

## 3.2 Implémentation

D'après la signature de notre processus $ETL$ ($\{(S_1, OL_1), ..., (S_n, OL_n), OG\} \rightarrow EDBO$), nous devons établir trois types d'implémentations concernant la récupération de l'ontologie locale de chaque source, la récupération de l'ontologie de l'$ED$ (l'ontologie globale) et finalement l'exécution des opérateurs génériques. Ces implémentations sont détaillées dans les sections suivantes.

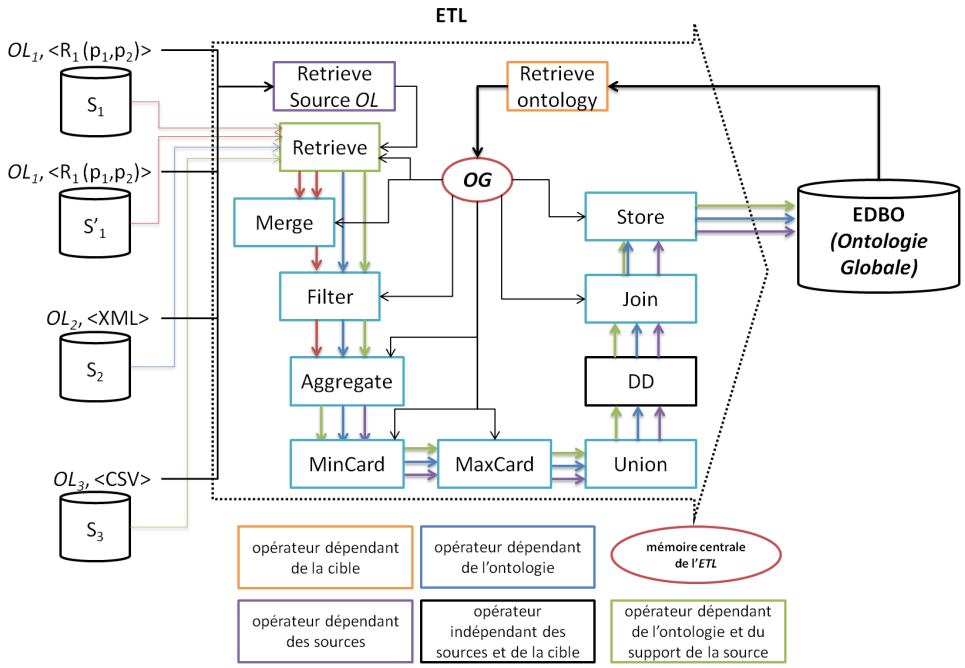


FIGURE 4.6 – Approche sémantique du processus *ETL*

### 3.2.1 Récupération de l'ontologie globale

Pour récupérer l'ontologie globale (*OG*), embarquée dans l'*EDBO*, le recours à un langage d'interrogation sémantique comme *SPARQL* [19] est nécessaire. Dans notre travail, nous avons utilisé le langage de requêtes *OntoQL* défini sur la plate-forme *OntoBD* (cf chapitre 2), et développé au sein du laboratoire LIAS.

Le langage *OntoQL* possède trois fonctionnalités [59].

- La définition des données, afin de créer les concepts et leurs propriétés en accord avec le méta-modèle d'ontologie.
- La manipulation des données, pour insérer, supprimer ou modifier des instances dans des tables.
- L'interrogation des données, pour exécuter des requêtes.

Afin de stocker les résultats qui vont être fournis par l'opérateur *Retrieve_ontology(EDBO)*, nous avons implémenté (en JAVA) dans l'*ETL* un modèle d'ontologie. Ce modèle contient tous les éléments ontologiques nécessaires aux opérateurs (classes et propriétés) (voir figure 4.7).

L'opérateur *Retrieve_ontology(EDBO)* commence par interroger l'*EDBO* sur les classes de l'*OG* :

```
SELECT #name,#oid FROM #Class
```

Cette requête indique que l'on souhaite récupérer le nom et l'*oid* (qui est une référence unique

19. SPARQL Protocol and RDF Query Language

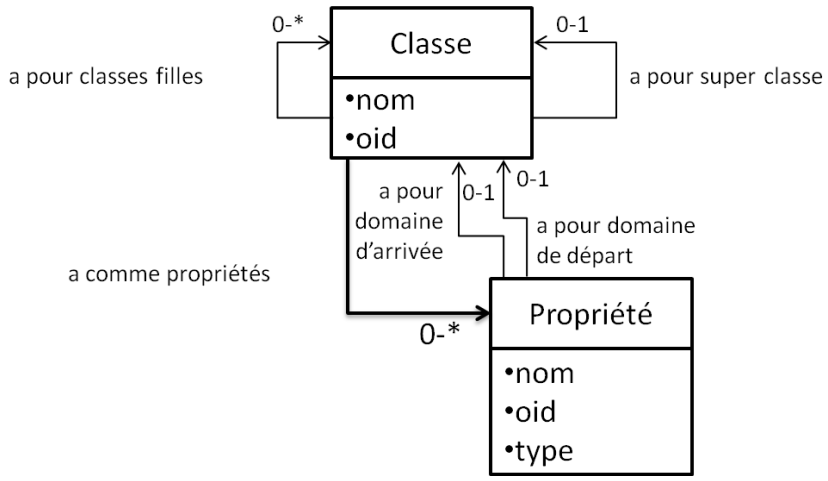


FIGURE 4.7 – Modèle d'ontologie de l'*ETL*

de chaque élément #Class) des classes. L'élément #Class contenu dans la partie *méta-schéma* de la plate-forme *OntoDB* permet de décrire les classes de l'*OG*. Ces classes vont être stockées dans le cache de l'*ETL* selon le modèle d'ontologie préalablement défini afin d'optimiser les accès. Ensuite, l'opérateur va compléter le modèle en récupérant les relations d'héritages entre les concepts. Ce point est fondamental étant donnée l'importance de l'héritage lors de la décomposition en briques de l'*OG*. Ainsi la requête suivante est exécutée pour chaque classe :

```
SELECT #directSuperclasses FROM #Class as c
WHERE c.#name='unNomDeClasse'
```

Cette requête recherche des super-classes directes d'une classe passée en paramètre. Si la requête aboutit à une réponse, la classe étudiée voit sa définition complétée (connaissance de la super-classe) et la description de la super-classe l'est également (connaissance d'une classe fille). A ce stade on dispose des classes et des relations d'héritages, l'action suivante est de récupérer les propriétés des classes. Pour chaque classe, l'opérateur va réaliser la requête suivante :

```
SELECT p.#name,p.#oid,
  CASE WHEN p.#range IS OF (REF(#StringType))     THEN 'String'
       WHEN p.#range IS OF (REF(#IntType))        THEN 'Int'
       WHEN p.#range IS OF (REF(#RefType))        THEN 'RefType'
       WHEN p.#range IS OF (REF(#CollectionType)) THEN 'Array'
       ELSE 'Unknown'
  END
FROM #Class AS c, UNNEST(c.#usedProperties) AS p
WHERE c.#name='unNomDeClasse'
```

Voici quelques explications sur cette requête.

- On cherche le nom et la référence (*oid*) des propriétés d'une classe.
- Suivant le domaine d'arrivée de la propriété on enregistre un mot clé particulier (*String, Int, RefType* (pour les relations) et *Array*). Si le domaine d'arrivée n'est pas identifié alors on renvoie un message (domaine d'arrivée *Unknown*).
- Ces propriétés sont cherchées pour la classe dont le nom est donné comme paramètre pour la requête (*unNomDeClasse*)

Les résultats viennent compléter la description des classes dans le modèle d'ontologie de l'*ETL*.

Enfin, lorsque qu'une propriété est une relation alors l'opérateur exécute une dernière opération avec la requête ci-dessous :

```
SELECT t.#onClass.#name
FROM #RefType AS t,
     #Class   AS c,
     UNNEST(c.#properties) AS p
WHERE p.#range = t.#oid
  AND c.#name='nomDeLaClasseDeDépart'
  AND p.#name='nomDeLaPropriété'
```

Cette opération vise à récupérer le nom de la classe d'arrivée de la propriété *nomDeLaPropriete*. Le *type référence* (#*RefType*) possèdent plusieurs attributs comme #*onClass* qui indique le domaine d'arrivée et *oid* qui est une référence unique. De plus, une propriété *p* qui correspond à une relation entre deux classes va posséder des attributs comme un nom (*p.#name*) et un domaine d'arrivée (*p.#range*). Parmi les références (#*RefType AS t*), les classes (#*Class AS c*) et les propriétés *p* des classes [20], on cherche la propriété avec le bon nom et la bonne classe de départ ainsi que la propriété dont le domaine d'arrivée correspond à une référence. Quand ces trois conditions sont réunies alors on connait le *type référence* de la propriété étudiée et on récupère le nom de la classe d'arrivée.

### 3.2.2 Récupération des ontologies locales

Comme indiqué plus tôt, les sources disposent chacune d'une *OL* qui les décrit. Dans les différentes expérimentations auxquelles EDF a participé, il a toujours été délicat d'accéder directement aux serveurs *SCADA* des différents partenaires. En revanche, nous avions accès à des extractions régulières d'une partie des données récupérées par ces serveurs sous la forme de fichiers *CSV*.

Nous avons donc implémenté l'opérateur capable de récupérer l'ontologie locale exploitée et contenue dans les fichiers *CSV*. Pour l'opérateur précédent (*Retrieve_ontology(EDBO)*), nous pouvions exploiter un langage de requêtes, mais nous avons opté pour une autre solution.

20. «L'opérateur *UNNEST* permet de transformer une collection en une relation pour qu'elle puisse être utilisée dans la clause *FROM*» [59]

Nous avons développé un programme JAVA pour récupérer la classe que représente un fichier grâce à son nom puis les propriétés des instances à l'aide des premières lignes du fichier.

On remarque que notre démarche nécessite le développement d'un opérateur en charge de la récupération de l'*OL*, non plus pour chaque source, mais pour chaque type de support des sources (dans notre cas nous avons développé un opérateur capable d'appréhender les fichiers *CSV*).

### 3.2.3 Implémentation des opérateurs du processus $ETL$

Les opérateurs précédemment définis au niveau sémantique ont été implémentés en JAVA à partir des spécifications que nous avons indiquées. La présence de l'ontologie au niveau du cache optimise les accès non nécessaires à l'*ED* et aux sources.

L'opérateur *Retrieve* doit être implémenté pour chaque support de stockage des sources de données (bases de données, fichiers *CSV*, etc.). L'hypothèse liée à notre projet est que la présence d'une ontologie globale référencée par l'ensemble des ontologies locales facilite l'implémentation de l'ensemble des opérateurs. Comme par exemple pour les opérateurs nécessitant de comparer des classes entre elles, l'une appartenant à une *OL* et l'autre à l'*OG*.

## 3.3 Synthèse

Avec l'approche classique nous aurions dû développer quasiment un $ETL$ par source de données (certains opérateurs pouvant être ré-utilisés comme la détection des doublons par exemple), ce qui aurait nécessité des ressources importantes (analyse des sources et développements logiciels). Mais surtout nous n'aurions pas exploité l'effort consenti dans la création de l'ontologie et sa diffusion.

Avec l'$ETL$ sémantique nous avons déplacé l'effort : au lieu de porter sur la connaissance de l'implémentation physique des sources et sur les développements celui-ci porte à présent sur la maintenance de l'ontologie. Il reste néanmoins une part d'implémentation spécifique à réaliser pour chaque support de sources (bases de données, fichiers *CSV, XML*, etc.) afin d'être capable de récupérer l'*OL* de chaque source. Toutefois cet effort est bien plus faible que celui nécessaire aux développements à réaliser dans l'approche classique sans ontologie.

La figure 4.8 résume le fonctionnement de l'$ETL$ sémantique : avec une partie manuelle pour maintenir les ontologies et une partie d'adaptation automatique.

Par rapport à la figure 4.5, la figure 4.6 permet de se rendre compte de l'évolution du processus $ETL$. On remarque au passage que certains opérateurs sont principalement liés aux sources et d'autres à l'*EDBO*.

Dernier point, grâce à l'$ETL$ sémantique l'*EDBO* peut être repeuplé après une évolution de l'ontologie, c'est-à-dire que l'*EDBO* avec l'ancienne version de l'ontologie devient une source

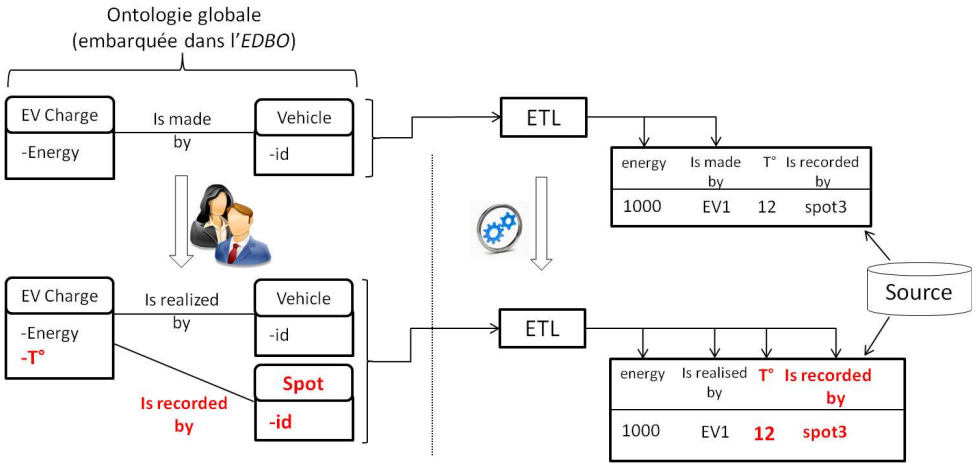


FIGURE 4.8 – *ETL* sémantique générique

de données. Toutefois certaines précautions sont à prendre. En effet, si de nouveaux éléments apparaissent dans le modèle de l'*EDBO* alors l'intégration de nouvelles données en tiendra compte mais ce ne sera pas nécessairement le cas pour les anciennes données.

Cas 1. *Les sources ne contiennent plus les anciennes données*. Alors les données doivent être récupérées dans l'ancienne version de l'*EDBO*, et dans ce cas les données seront nécessairement incomplètes. Il faut donc que l'*ETL* mette en place deux mécanismes :
- L'ontologie doit spécifier quelle valeur donner aux champs manquants.
- L'*ETL* doit ajouter à la donnée la version de l'ontologie qui correspond à sa validité.

Cas 2. *Les sources contiennent les anciennes données et les données sont incomplètes*. Cette situation se ramène au premier cas.

Cas 3. *Les sources contiennent les anciennes données et les données sont complètes*. C'est le cas idéal, les données sont chargées et l'*ETL* indique que ces données sont conformes à la dernière version de l'ontologie.

Le versionnage et les valeurs à indiquer par défaut des données permettent de maintenir la qualité de l'*EDBO*. Ces mécanismes permettent aux experts d'affiner leurs requêtes pour que ces dernières ne soient pas affectées par les changements dans l'ontologie.

# 4 Implémentation et résultats

## 4.1 Besoins

Pour implémenter notre solution nous avions besoin d'une plate-forme capable de gérer une ontologie et ces instances. La plate-forme doit également supporter une base de données afin d'y créer notre *ED*, et proposer un langage de requête permettant de manipuler chacun des éléments. Il existe aujourd'hui de nombreux *SGBD* correspondant à ces besoins (Oracle [91],

*KAON* [18], etc.). Pour nos travaux nous avons utilisé la plate-forme *OntoDB* [31].

## 4.2 *OntoDB* et le langage *OntoQL*

La plate-forme *OntoDB* développée au LIAS [21] répond à ces exigences. Elle est basée sur le *SGBD* PostgreSQL et permet de mettre en place les concepts des bases de données à base ontologique (*BDBO*) [99].

Pour permettre *de gérer à la fois des ontologies et des données*. La plate-forme est constituée de quatre parties (voir figure 4.9) :

1. *La partie méta-schéma*. Cette partie stocke le modèle d'ontologie utiliser pour définir une ontologie. C'est dans cette partie que l'on définit ce qu'est un concept ontologique, il possède : un nom et une définition - déclinés en français et en anglais -, une référence unique et des propriétés. Une propriété possède les mêmes éléments de référencement, de nommage et de définition qu'un concept auxquels s'ajoute le domaine d'arrivée. Cela peut être un type (entier, string, etc.) ou un concept ontologique (une référence). Enfin une ontologie est un ensemble de concepts ontologiques.
2. *La partie ontologie*. Dans notre cas il s'agit de l'ontologie qui va être une instance du modèle d'ontologie stocké dans la partie méta-schéma. C'est dans cette partie qu'est stockée l'ontologie de la *ME*, la définition d'une brique ontologique et des concepts avec leurs propriétés.
3. *La partie catalogue système*. Cette partie est prédéfinie dans tous les *SGBD* et contient : la description des éléments de la base données : table, indexe, clés primaires, etc.
4. *La partie donnée*. Les tables de cette partie correspondent aux concepts ontologiques sélectionnés par les utilisateurs et les données stockées dans ces tables sont les instances des différents concepts. Dans notre cas nous aurons, par exemple, une table «*EvOwner*» pour les possesseurs de $\mathcal{VE}$, «*EV charge*» pour les charges des $\mathcal{VE}$, etc.

Ces quatre parties sont liées : l'*ontologie* respecte le *modèle d'ontologie* défini dans la *partie méta-schéma*. Les *instances* stockées dans la *partie données* correspondent aux concepts définis dans la partie ontologie. Enfin, les *tables* utilisées pour cette partie sont stockées dans le *catalogue système*. *OntoDB* permet d'ajouter des concepts ontologiques qui seront représentés comme des tables et de choisir les propriétés à utiliser. Cette capacité est fondamentale pour notre solution, en effet la plate-forme prend complètement en charge la création des tables et de leurs paramètres (noms des tables et des champs, typages, clés primaires et étrangères, etc.). De fait, l'ontologie sert d'interface entre l'expert et le système d'information dans la mesure où la plate-forme prend en charge la partie *technique*.

Pour réaliser ces opérations la plate-forme *OntoDB* met à disposition le langage *OntoQL* précédemment décrit. Des exemples de requêtes sont donnés dans la suite de ce chapitre pour

21. Laboratoire d'Informatique et d'Automatique des Systèmes

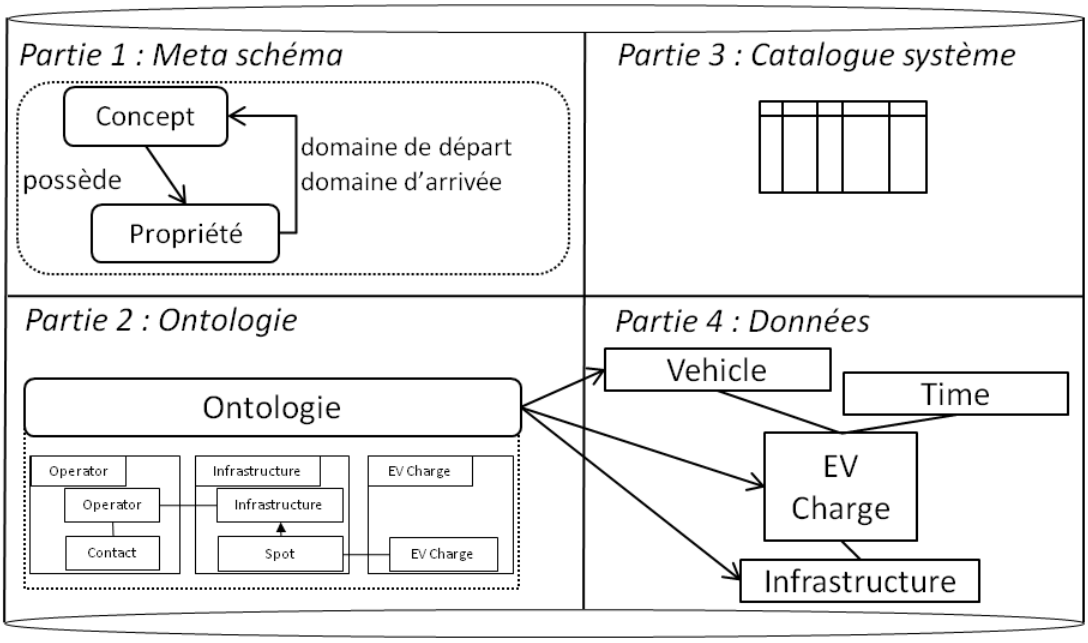


FIGURE 4.9 – Plate-forme *OntoDB*

illustrer l'implémentation de l'*EDBO*.

## 4.3 Entrepôt de données à base ontologique de la *ME*

La première étape afin de mettre en place l'*EDBO* consiste à définir l'ontologie de la *ME* dans la plate-forme *OntoDB*. Des exemples des requêtes *OntoQL* utilisées sont présentées dans cette section.

### 4.3.1 Schéma de l'*EDBO* sur la *ME*

L'application des démarches de définition des besoins et de création des différents modèles (conceptuel, logique et physique) à partir de l'ontologie définie dans le chapitre précédent a permis de concevoir l'*ED* de la *ME*. La figure 4.10 montre le schéma conceptuel obtenu en prenant comme **fait la charge d'un** $\mathcal{VE}$.

Ce schéma comporte cinq dimensions :

- *La dimension temporelle* permet d'historiser les charges réalisées par les $\mathcal{VE}$ que ce soit pour des questions de facturation ou d'analyse de l'évolution des habitudes.
- *Le type de charge* est un indicateur important, il peut s'agir d'une charge normale (c'est-à-dire avec une puissance appelée de 3 kW et qui va durer 6 à 8 heures) ou d'une charge rapide (d'environ 30 minutes) requérant davantage de puissance. Le type de charge est à la fois une demande de l'utilisateur en un lieu et un moment donné et également une contrainte particulière sur le réseau de distribution d'électricité.

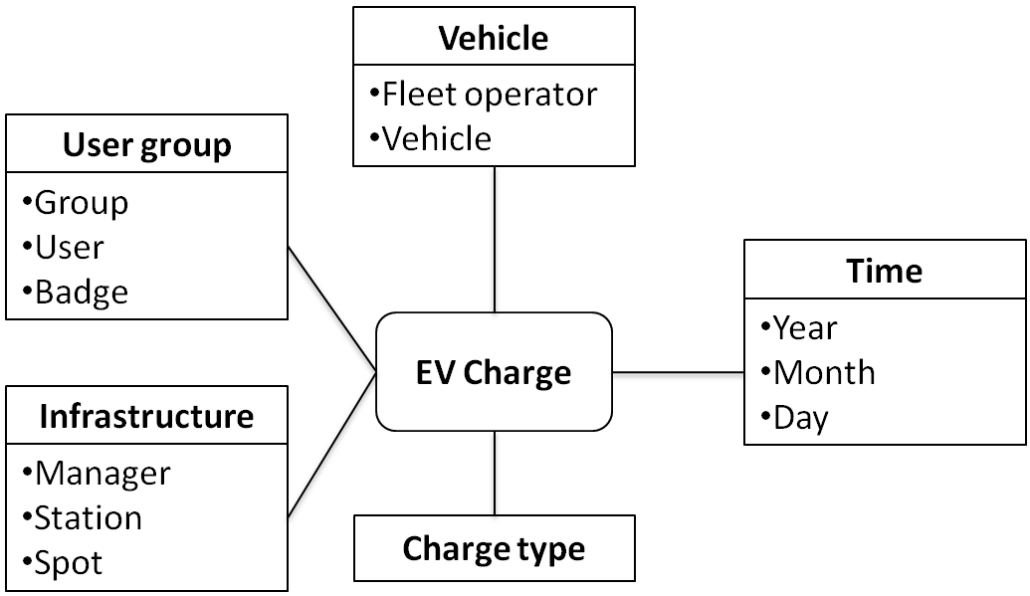


FIGURE 4.10 – Schéma conceptuel en étoile de l'entrepôt

– *Les infrastructures* permettent de localiser les charges dans l'espace. D'un point de vue abstrait par rapport au réseau de distribution de l'électricité mais également d'un point de vue géographique par rapport aux trajets des utilisateurs et des offres de charge disponible (super-marché, concessionnaires, etc.).
– *Les groupes d'utilisateurs* correspondent à la typologie des utilisateurs. Cette dimension correspond aux études de marchés et aux études sociologiques menées par EDF. Elle sera plus amplement décrite dans le chapitre suivant.
– *Les véhicules*, il est apparu utile de pouvoir analyser les charges du point de vue des véhicules. Notamment pour établir des corrélations avec les autres dimensions et pour pouvoir rapporter l'activité des véhicules aux gestionnaires de flotte.

### 4.3.2 Quelques éléments de métrologie

L'ontologie comporte aujourd'hui 31 concepts et plus de 118 propriétés. Parmi ces éléments 14 concepts ont été sélectionnés pour composer le modèle conceptuel et 34 propriétés de ces concepts figurent dans l'*EDBO* déployé. Le but de l'*EDBO* est, pour le moment, de répondre aux besoins émis par le projet $\mathcal{VE}$ d'EDF R&D mais l'évolution du domaine laisse présager un usage plus étendu pour gérer différents services.

### 4.3.3 Construction du schéma de l'*EDBO* avec *OntoQL*

**Définition d'un concept.** La figure 4.11 illustre la requête effectuée pour créer un concept, en l'occurrence celui d'une borne de charge. La description du concept doit respecter le modèle

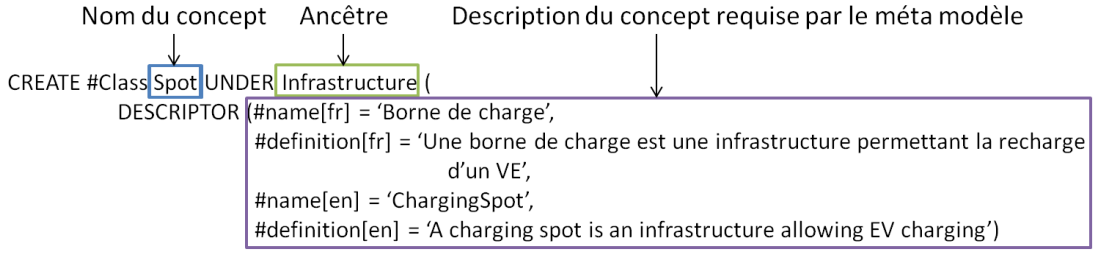


FIGURE 4.11 – Création d'un concept

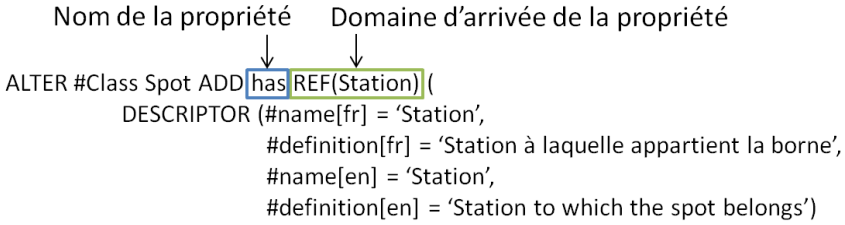


FIGURE 4.12 – Création d'une propriété

d'ontologie dont dispose la plate-forme. Ici le modèle impose de donner un nom et une description du concept en français et en anglais. A l'avenir d'autres informations pouvant être requises, comme d'autres langues ou une référence au versionnage.

Pour compléter les concepts l'opération consiste à modifier le concept pour lui ajouter une propriété. Il faut préciser le nom de la propriété et son domaine d'arrivée : une référence pour une relation ou un type (chaîne de caractères, entiers, etc.). La figure 4.12 illustre la requête à effectuer.

Les deux requêtes présentées permettent d'intégrer l'ontologie à la plate-forme dans la partie correspondante. La figure 4.13 montre la requête permettant de choisir les concepts de l'ontologie qui seront utilisés pour définir les instances. Il est également nécessaire de préciser les propriétés qui sont utilisées pour caractériser les instances.

**Définition des instances.** Voici succinctement comment sont effectuées les opérations de bases sur les instances :

– *Create* : INSERT INTO Spot VALUES ("un nom", stationRueX, "borne de charge normale/rapide", "manuelle", "charge normale"). Pour utiliser la table créée par l'extension de

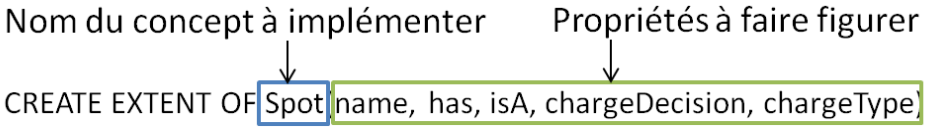


FIGURE 4.13 – Extension (implémentation) du concept dans la partie données

la classe Spot à la partie donnée on utilise le terme ontologique, la plate-forme se charge du reste.
- *Read* : SELECT * FROM Spot.
- *Update* : UPDATE Spot WHERE has = stationRueX SET name = "nouveau nom".
- *Delete* : DELETE Spot WHERE name = "nouveau nom".

La syntaxe d'*OntoQL* pour le traitement des données est proche de celle de *SQL*, dans tous les cas les opérations sont réalisées à partir des termes de l'ontologie car la plate-forme se charge de faire les liens vers les tables et les attributs correspondants.

# 5 Conclusion

Nous avons vu dans ce chapitre comment la méthode modulaire et incrémentale de création d'une ontologie permet d'assister la démarche de conception, de déploiement et de peuplement d'un $EDBO$. En effet le découpage de l'ontologie en briques et la connexion des briques selon des contraintes précises permet d'accélérer le processus de création des différents modèles de l'$EDBO$. La flexibilité de la méthode de construction de l'ontologie se retrouve lors de la création du modèle conceptuel qui peut facilement évoluer.

Afin de peupler l'$EDBO$, l'$ETL$ sémantique assure le chargement des données des sources vers l'$EDBO$ aussi bien que d'une version de l'$EDBO$ à une autre. Grâce à ceci il est possible d'accompagner les évolutions de l'ontologie et du modèle de données de l'$EDBO$ qui en dépend sans avoir à re-développer un, ou des, $ETL$. Ce point constitue l'un des arguments important pour justifier les travaux auprès d'un industriel. Le gain est directement mesurable en tant que coût évité.

Il existe depuis plusieurs années des $SGBD$ permettant de manipuler des ontologies et des données. Ces plates-formes, comme la plate-forme *OntoDB*, ont ouvert la voie à des travaux et des tests plus poussés sur les ontologies. La plate-forme *OntoDB* propose ainsi, grâce au langage *OntoQL*, une grande liberté dans la manipulation des ontologies, des tables les représentant et des données associées.

Les capacités du langage *OntoQL* ont permis de mettre en place rapidement l'$EDBO$ à partir de l'ontologie en sélectionnant des concepts pour former le modèle conceptuel. Ce même langage, manipulé cette fois par l'$ETL$, a permis de peupler l'$EDBO$ et de permettre son exploitation.

Dans l'état l'$EDBO$, offre aux utilisateurs **des possibilités d'analyses et de *reporting*** en se basant uniquement sur l'ontologie pour, à la fois, **intégrer des données et les interroger**. L'effort à fournir pour intégrer une nouvelle source est minime : il s'agit de mettre à jour l'ontologie. L'ontologie devient alors l'interface privilégiée entre l'$EDBO$ et les utilisateurs. Les utilisateurs peuvent donc maintenant se focaliser sur l'exploitation des données et mettre à profit le temps gagné pour produire des résultats à partir de l'$EDBO$.

Une partie de l'objectif initial est donc remplie. **Les utilisateurs gagnent du temps** pour exploiter les données et en **tirer des résultats pour l'aspect *affaires* du projet**. La valeur des experts se situe précisément dans cette exploitation à des fins commerciales des données. Le prochain chapitre s'attache à cette notion en proposant une automatisation des traitements afin que les experts puissent se consacrer au maximum sur l'aspect *valeur ajoutée* généré par leur travail.

Chapitre 5

# Gestion de la connaissance et processus métiers

## Sommaire

# 1 Introduction

Le premier objectif de ces travaux était de fournir à EDF une plate-forme pour intégrer les données et d'en assurer la qualité puis de les stocker à des fins d'analyses. Cette objectif a été atteint avec la plate-forme présentée dans le chapitre précédent.

La plate-forme accueille un entrepôt de données ($ED$) déployé à partir de l'ontologie, l'association d'une ontologie et d'un $ED$ forme un entrepôt de données à base ontologique ($EDBO$). L'ontologie est ensuite exploitée par un $ETL$ sémantique, spécifiquement conçu dans le cadre de ces travaux, pour peupler l'$EDBO$ en s'adaptant automatiquement aux changements dans l'ontologie et à leurs répercussions sur l'$EDBO$. Ainsi nous disposons à présent d'une forte capacité d'intégration de données hétérogènes et de leur stockage à partir d'un élément descriptif : l'ontologie de la $ME$.

Grâce aux travaux précédents on peut maintenant revenir à la problématique initiale : traiter la mine de processus disponibles. C'est-à-dire de gérer les connaissances (pour la prise de décisions et les rapports) comme on gère les données d'un $ED$. Jusqu'à présent, la communauté $ED$ a organisé son travail sur les données et ces travaux ont mené aux développements de techniques d'optimisation du stockage en travaillant sur le cycle de création de l'$ED$. Ces travaux vont de l'intégration des besoins des décideurs dès les premières étapes du cycle de conception à l'optimisation du schéma de l'$ED$ par rapport aux requêtes. D'autre travaux portent sur l'optimisation des requêtes vis-à-vis de la masse de données à traiter. Pour tirer parti et pour compléter ces travaux nous nous sommes attachés à la connaissance, *i.e.* la gestion des résultats des requêtes et des calculs dont ils font l'objet. Avec cette dernière étape le travail présenté propose de couvrir tout le cycle de vie d'une donnée, depuis son intégration jusqu'à son exploitation pour des processus d'aide à la prise de décisions, à partir d'éléments déclaratifs : les ontologies.

Afin de poursuivre notre démarche, les connaissances ont été formalisées au sein d'une ontologie pour former l'*ontologie des connaissances* aussi appelée *ontologie métier*. La description des connaissances est enrichie en reliant les concepts des connaissances aux concepts de l'ontologie de domaine. Les instances de ces connaissances sont le résultat de requêtes et de calculs réalisés sur les données, ces opérations sont appelées ici des **processus métiers**. Pour rendre la démarche de formalisation des connaissances consistante, l'ontologie sera appuyée par la modélisation des processus avec la *Business Process Modelisation and Notation* ($BPMN$), c'est une représentation visuelle formalisée des éléments d'un processus (début, fin, tâches, connecteurs, etc.). Nous proposons ici une intégration au niveau sémantique des processus métiers.

Tous les éléments de notre solution étant posés, ce chapitre présente un exemple de leur implémentation dans un système de gestion de base de données ($SGBD$). Pour illustrer le fonctionnement de la solution décrite ce chapitre propose une série d'exemples de la mobilité électrique ($ME$).

# 2 Ontologie des connaissances

## 2.1 Définition et intérêt

Telle que nous l'avons définie, l'ontologie sur la $ME$ décrit le domaine (les équipements, les acteurs et les données). Dans ce chapitre les concepts qui nous intéressent portent sur l'usage des $\mathcal{VE}$, ils représentent les *connaissances métiers*, ou **processus métiers**, d'EDF. Ces connaissances sont composées de deux éléments :

- Il y a d'abord la description de la connaissance. Cette description comprend tous les éléments nécessaire pour la comprendre : à qui elle s'adresse, pourquoi elle est pertinente, quelle elle est sa forme, quel est son format, etc.
- Ensuite il y le processus pour la calculer et l'indication des données nécessaires au processus.

Ces connaissances viennent formaliser sous la forme d'une ontologie la mine de processus à exploiter, dans notre cas la mine correspond aux connaissances des différents experts d'EDF. A ce titre il y a une distinction importante entre les deux ontologies : **l'ontologie de domaine est publique et diffusée** afin d'être exploitée par un maximum de partenaires, **l'ontologie des connaissances, en revanche, est propre à EDF** car elle représente de la valeur ajoutée par la R&D d'EDF aux données et contribue aux activités commerciales d'EDF.

Une connaissance métier est le fruit des travaux et des analyses sur le domaine menés par des experts. A partir des connaissances métiers les experts peuvent prendre des décisions, réaliser des modèles de prévisions, etc. Il est donc primordial de rendre les connaissances facilement accessibles. Si la description du domaine permettait de gagner du temps, la mise à disposition des connaissances permet aux experts de consacrer plus de temps à exercer leur métier plutôt qu'à réunir les données. A titre d'exemple une connaissance métier peut être :

- Un indicateur simple : la fréquence d'utilisation d'une borne de recharge.
- Des informations agrégées : la courbe de charge d'un utilisateur (la définition d'une courbe de charges est donnée dans ce chapitre).
- Des indicateurs complexes : algorithmes d'apprentissage (*machine learning*) pour grouper des profils d'utilisateurs.
- etc.

Chacune de ces connaissances possède une définition, des requêtes associées (sur les données du domaine ou sur d'autres connaissances), des méthodes de calcul, etc. Il existe donc une formalisation des connaissances mais qui est propre à chaque expert. Elle n'est ni mutualisée entre les experts et les équipes, ni capitalisée pour les prochains experts.

A partir de ce constat, nous avons établi notre approche des connaissances comme s'il s'agissait d'un domaine à décrire. Cela nous a amenés à créer une seconde ontologie, l'*ontologie des connaissances*, dont les concepts reprennent la formalisation précédente. Afin de compléter cette ontologie chaque connaissance est reliée aux concepts de l'ontologie de domaine nécessaires.

Le résultat de cette démarche offre de nouvelles facilités pour manipuler les données :

1. Il n'y a plus de travail redondant, ce qui représente un gain de temps. En effet les experts ont à leur disposition le contenu des travaux menés : les indicateurs, les méthodes de calcul, etc. La formalisation des connaissances permet de les ré-utiliser plus facilement.
2. Il est possible de voir pour chaque élément du domaine les connaissances qui s'y rattachent. Il devient possible de voir quelles sont les connaissances liées à un élément du domaine en parcourant les ontologies.
3. Chaque connaissance est reliée à son destinataire ou à un projet. Il est possible de spécifier un destinataire à une connaissance, cela permet d'accélérer la production des rapports.
4. Il est possible de versionner la définition d'une connaissance, dans un souci de conserver les méthodes de calculs employées pour les anciennes connaissances.
5. La transmission de la connaissance est assurée, entre experts ou entre projets.

La construction de l'ontologie des connaissances, ou ontologie métier, s'effectue selon la méthode décrite dans le chapitre 3. Lors de la création de l'ontologie du domaine nous avons mentionné que les experts avaient formulé leurs besoins souvent sous la forme «il me faut cette information». A partir de là nous avons procédé à l'étape de *spécifiaction* de l'ontologie. La suite de la formulation citée était souvent «pour réaliser telle et telle analyses». Ainsi en spécifiant l'ontologie de domaine les experts ont également commencé à spécifier les processus qu'ils souhaitaient mettre en place sur le domaine. Nous avons donc exploité le travail déjà réaliser et nous l'avons complété avec les experts pour aboutir à l'ontologie des processus métiers qui caractérise la mine de processus identifiée dans l'introduction.

## 2.2 L'ontologie des connaissances d'EDF sur la ME

Cette section présente plusieurs connaissances exploitées par EDF sur la $ME$, ces exemples permettent de mettre en avant plusieurs niveaux de complexité d'élaboration. Nous suivrons ces mêmes exemples dans les sections suivantes pour illustrer ce chapitre. Dans les exemples nous classons les connaissances par niveaux (voir figure 5.1) :

- Le niveau 1 contient les connaissances générées directement à partir des données du domaine (décrites par l'ontologie de domaine).
- Le niveau 2 contient les connaissances générées à partir de données du domaine et des autres connaissances (niveau 1, 2 ou 3).
- Le niveau 3 regroupe les connaissances calculées sur les autres connaissances de tous niveaux mais pas sur des informations du domaine.

Les paragraphes ci-dessous illustrent ces différents niveaux de connaissances et comment on progresse depuis le premier niveau jusqu'au dernier en gagnant en abstraction par rapport aux données du domaine.

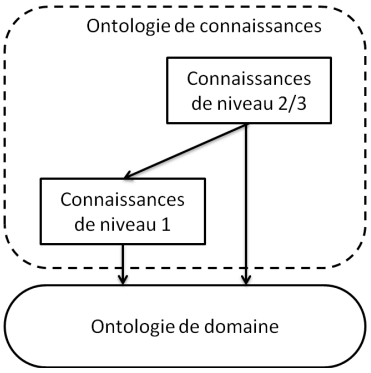


FIGURE 5.1 – Le niveau 1 comprend les connaissances basées sur le domaine, le niveau 2 contient les connaissances s'appuyant sur le domaine et les autres connaissances et le niveau 3 correspond aux connaissances s'appuyant exclusivement sur les connaissances des niveaux 1 et 2.

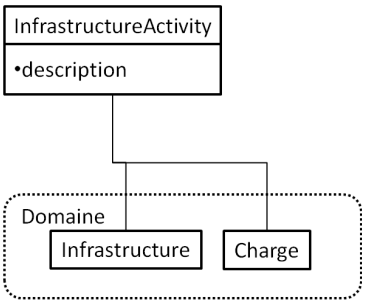


FIGURE 5.2 – Activité des infrastructures

### 2.2.1 Activité des infrastructures

Afin d'assurer un service de qualité, des données relatives aux infrastructures de recharges sont nécessaires (voir figure 5.2). Pour limiter les équipements à installer sur une infrastructure et donc de diminuer leurs coûts il est intéressant de déduire l'état de l'infrastructure à travers l'usage qui en est fait. Cette connaissance à vocation à détecter les cas suivants :

- Une baisse marquée de l'usage, ceci peut correspondre à une panne d'un équipement et requiert une intervention sur le terrain. Dans le cas où il s'agit d'une borne au sein d'une station une baisse soudaine est probablement liée à une panne, s'il s'agit de toute la station alors il peut s'agir d'un problème d'alimentation ou de problèmes de transmission de données.
- Une activité faible par rapport aux prévisions ou aux infrastructures à proximité amène des questions relatives à l'accès, aux offres qui y sont disponibles ou à l'emplacement des installations.
- De la même manière l'augmentation de l'activité doit aussi être signalée afin de garantir l'accès à la recharge.

L'activité des infrastructures font partie du premier niveau de connaissances, c'est-à-dire qu'elles se basent uniquement sur les données du domaine.

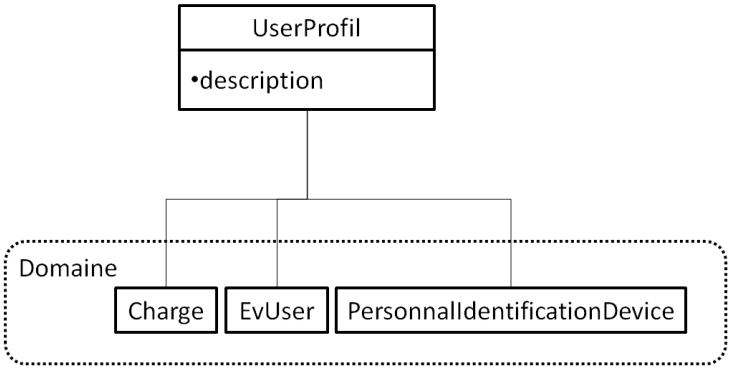


FIGURE 5.3 – Profil utilisateur

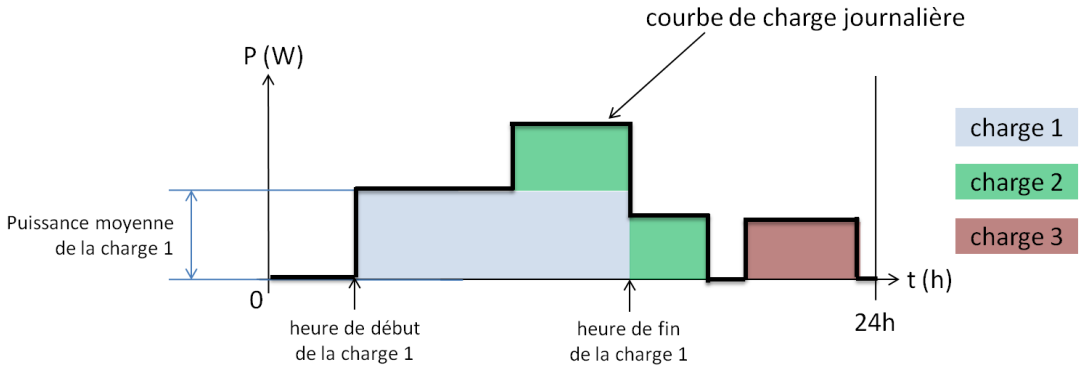


FIGURE 5.4 – Calcul de la courbe de charge (journalière)

### 2.2.2 Profil utilisateur

Il s'agit de l'indicateur privilégié par EDF pour réaliser des analyses sur les utilisateurs de $\mathcal{VE}$ (voir figure 5.3), le profil est caractérisé par un identifiant et sa courbe de charge. La courbe de charge représente la consommation moyenne sur une journée : il s'agit d'une courbe représentant la puissance appelée sur 24h. Pour la calculer il faut agréger les charges et calculer la puissance appelée par chaque charge. La figure 5.4 montre comment la courbe est obtenue :

- Il faut récupérer les charges et leurs heures de début et de fin, où l'heure est égale au reste de la division euclidienne de la date complète par la période étudiée.
- Pour chaque pas de temps on calcule la somme des puissances (moyennes) des charges qui se déroulent sur ce pas de temps.
- Le résultat est un tableau qui à chaque pas de temps associe une puissance.

Tout comme l'activité des infrastructures, il s'agit une connaissance du premier niveau.

Pour illustrer cette connaissance la figure 5.5 présente la courbe de charge journalière (normalisée par son maximum) d'un utilisateur. Á la lecture de ce type de connaissances on va remarquer que l'utilisateur se charge à deux moments particuliers : le matin à partir de 8h et en début d'après-midi, autour de 14h. Les charges du matin sont caractéristiques des trajets

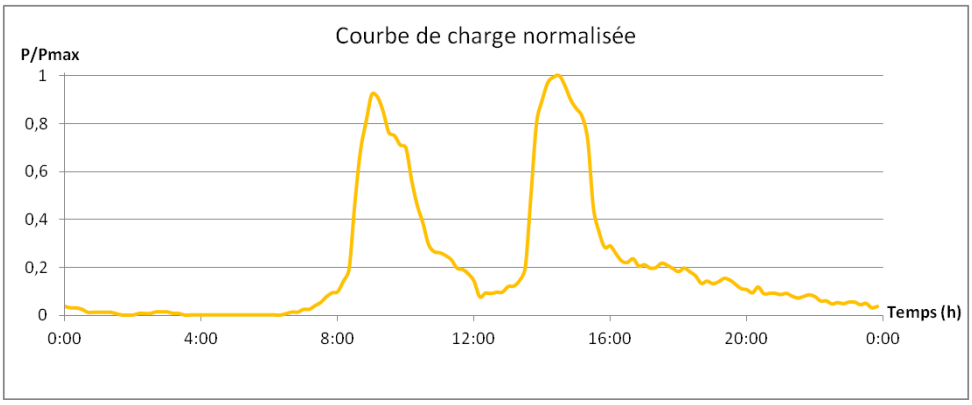


FIGURE 5.5 – Courbe de charge journalière, normalisée par la puissance maximale relevée, d'un utilisateur

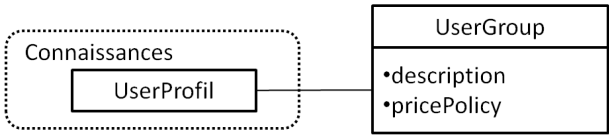


FIGURE 5.6 – Groupe d'utilisateurs

domicile-travail et compte-tenu de la similitude des deux moments de charge, même hauteur et même largeur, on peut supposer que cet utilisateur fait un trajet équivalent à midi dans la mesure où l'énergie consommée, et donc rechargée, est proportionnelle à la distance. Ce type d'analyse est détaillé dans la suite du chapitre.

### 2.2.3 Groupe d'utilisateurs

Il n'est pas raisonnable d'envisager le traitement des utilisateurs au cas par cas au moment d'établir des contrats de fourniture d'électricité ou de proposer des offres particulières. Cela nécessiterait un suivi individuel important et il faudrait une connaissance extrêmement juste. Pour s'affranchir de ces difficultés EDF travaille sur des groupes d'utilisateurs.

Les groupes (voir figure 5.6) sont formés automatiquement à partir des profils utilisateurs décrits ci-dessus. Pour regrouper les profils des algorithmes d'apprentissage sont exploités pour former des groupes homogènes. Ces calculs sont détaillés dans le chapitre 6 en voici quelques éléments. La division des utilisateurs en des groupes de plus en plus réduit dépend de l'homogénéité du groupe. On va chercher à diviser les principaux groupes identifiés par les algorithmes d'apprentissage tant que les nouveaux groupes obtenus sont homogènes. L'homogénéité d'un groupe se définit par l'écart entre les prévisions faites pour ce groupe et la réalisation.

Ainsi, un groupe homogène contient des utilisateurs avec des profils similaires dont on peut prévoir avec un certain niveau de confiance les actions et les réactions. Cela permet de faire des contrats et des offres précises et ajustées à la consommation du groupe.

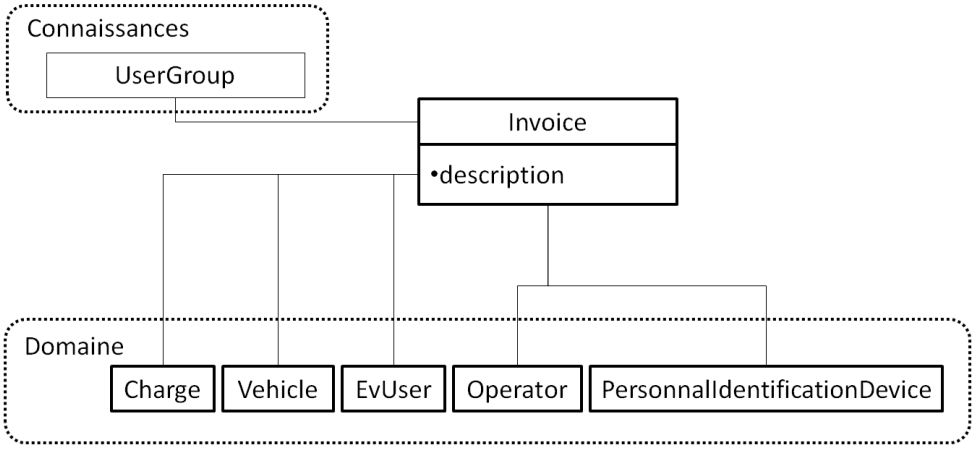


FIGURE 5.7 – Facture

La génération de cette connaissance s'appuie sur d'autres connaissances et non sur les données du domaine, c'est donc une connaissance de niveau trois.

### 2.2.4 Facturation

Il s'agit d'un des rapports les plus générés. Une facture (cf figure 5.7) doit regrouper les données d'un client (que ce soit un gestionnaire de flotte ou un individu), sa consommation sur une période donnée et son tarif. Le montant à régler par le client est déterminé à partir de ces informations, ainsi la partie description va contenir le calcul du montant à régler en fonction des informations sur le client et de ses charges.

On remarque que cette connaissance, bien que simple, requiert des données du domaine et des connaissances, c'est donc une connaissance du niveau 2.

### 2.2.5 Analyse des éléments temporels

Pour rapporter l'activité et les habitudes des groupes on analyse les données temporelles. Plusieurs rapports sont requis : les courbes de charge, les histogrammes représentant la durée des charges en fonction de leurs fréquences, etc. Ces connaissances peuvent relever des différents niveaux, elles sont détaillées dans le chapitre suivant (chapitre 6).

## 3 Processus métiers

La formalisation des connaissances permet aux experts de gagner du temps, et fournit de plus un système d'informations pour capitaliser les connaissances. Toutefois, l'ontologie des connaissances n'est jamais qu'une description des connaissances et de leur «mode d'emploi».

Les travaux des communautés de recherche proposent des techniques de pointe pour créer, gérer et optimiser l'utilisation d'un *ED*. Ces travaux laissent aux utilisateurs la mise en place de leurs applications et de leurs processus métiers (*PM*). Nous proposons d'intégrer ces processus au niveau sémantique, en formant une extension exécutable de l'ontologie des connaissances.

Cette extension des connaissances reprend le concept présent dans la plate-forme *OntoDB* qui consiste à étendre une ontologie en une base de données ou en un entrepôt de données.

## 3.1 Définition et intérêts

Les *PM* sont le chaînon manquant entre les données du domaine contenues dans l'entrepôt de données à base ontologique et les processus métiers. Jusqu'à présent ceux-ci étaient exécutés par les experts de façon décentralisée. Avec l'ontologie des connaissances il était possible pour un expert de s'approprier des analyses mises au point par d'autres experts. Toutefois cet expert devait implémenter les calculs nécessaires pour mettre en pratique ces analyses, or une telle implémentation est purement technique (écriture des requêtes et traitement des résultats). Il paraît donc intéressant d'automatiser cette étape d'implémentation afin de libérer les experts de cette tâche.

La définition d'un *PM* prolonge donc celle des connaissances. La connaissance, telle que représentée dans l'ontologie, est purement descriptive et un *PM* permet de la réaliser. *Un processus métier correspond à une modélisation des opérations à réaliser pour générer une instance d'une connaissance.*

Les opérations d'un *PM* se décomposent en plusieurs types :

1. Les requêtes : elles portent sur le domaine ou sur les connaissances.

2. Les tâches : ce sont des traitements à réaliser sur des données issues de requêtes. Il peut s'agir de calcul à mener ou de rapports à envoyer à différents destinataires.

3. Les choix : suivant les résultats des opérations précédentes la suite des opérations peut être soumise à des conditions (valeur limite, cas particulier, etc.).

Pour EDF c'est un point crucial, aujourd'hui les données sont traitées par différents experts de manière isolée et parfois répétitive d'un jeu de données à un autre. L'*ED* de domaine permet de centraliser les données afin de travailler moins à l'intégration des données, l'étape suivante est de faire la même opération pour les processus métiers existants. C'est cette nouvelle étape qui va permettre de travailler plus sur l'analyse et donc la valeur ajoutée par les experts d'EDF aux données recueillies.

## 3.2 Du cycle de vie de l'ontologie au cycle de vie des processus

### 3.2.1 Cycle de vie des processus

Le cycle de vie des processus métiers [9, 145] est proche du cycle itératif pour les ontologies [66], il se décompose en 5 principales étapes :

1. *Conception* : il faut définir les processus existants et identifier les procédures qui les composent. Ce travail s'effectue avec les acteurs de ces processus pour aboutir à une conception théorique et plus performante des processus existants.
2. *Modélisation* : suite à la conception, la phase de modélisation peut démarrer. Au cours de cette phase il faut répertorier les différents scénarios auxquels les processus vont être confrontés (*what-if analysis*) et décrire les processus de façon analytique.
3. *Exécution* : cette étape va devoir automatiser les processus, c'est une phase technique qui va s'appuyer sur la modélisation des processus.
4. *Suivi* : suivant leur importance les processus vont être suivis afin de vérifier leurs bonnes exécutions.
5. *Optimisation* : cette dernière étape du cycle de vie vise à améliorer les processus à partir des observations faites lors de la phase de suivi. Cette dernière phase renvoie vers la première si des modification sont à faire.

### 3.2.2 Interaction du cycle de vie de l'ontologie et de celui des processus

Dans notre démarche les processus génèrent les connaissances métiers décrites par l'ontologie. C'est pourquoi nous avons enrichi la phase de suivi en ajoutant l'enregistrement des exécutions à cette phase. C'est cette phase qui va conduire à la génération des connaissances pour les experts. Comme les processus génèrent les connaissances sur un domaine qui évolue, les processus peuvent changer pour générer de la connaissance. L'étude des changements à réaliser fait partie de la phase d'optimisation. En cas de changement, l'ancienne version du processus est historisée afin de conserver la mémoire de processus ayant permis la génération des anciennes connaissances.

La figure 5.8 montre l'interaction entre le cycle de vie de l'ontologie des connaissances et le cycle de vie des processus métiers :

- *Conception* : la phase de conception des processus métiers découle directement de l'ontologie. En effet, la formation de celle-ci est basée sur la description des processus, leurs buts et les acteurs impliqués. On va donc retrouver dans l'ontologie la description du processus en langage naturel : les tâches, les paramètres et la signature du processus.
- *Modélisation* : l'ontologie des processus métiers (ou des connaissances) a été la première étape pour décrire et partager les processus métiers entre les experts (capitalisation, historique, transmission, etc). La conceptualisation des processus dans l'ontologie apporte

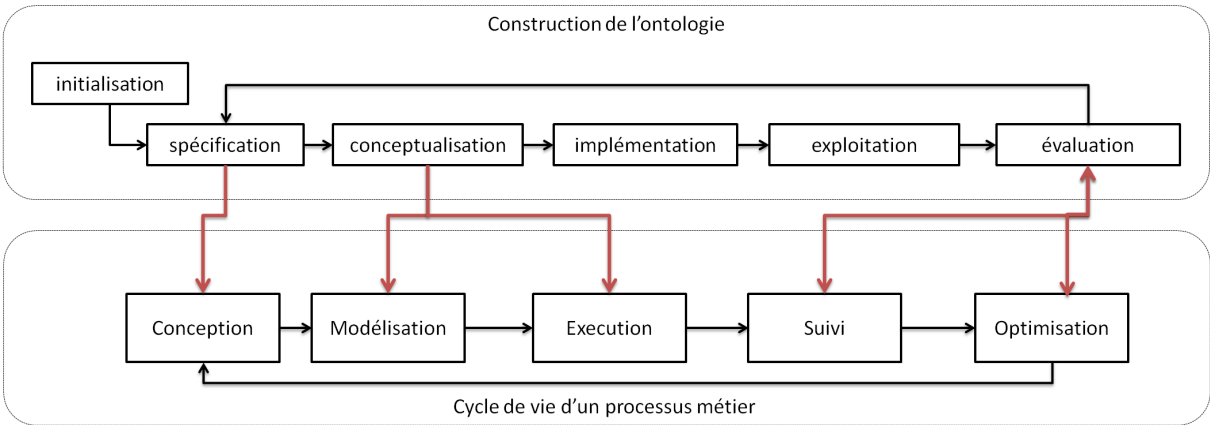


FIGURE 5.8 – Interaction entre le cycle de vie de l'ontologie et celui d'un processus métier

des éléments précis exploitables pour modéliser le processus : définitions des entrées et des sorties, etc, c'est-à-dire une description analytique des processus. La phase de modélisation est traitée plus en détail dans les paragraphes suivants avec l'exploitation d'un formalisme.

– *Exécution et suivi* : l'ontologie décrit comment fonctionne les processus et également ce qu'il en est attendu : envoi de rapports, enregistrement dans un entrepôt, etc. Le suivi des exécutions se traduit par l'enregistrement des exécutions (et de leurs résultats) dans un entrepôt. Cela permet de réaliser un suivi précis des exécutions et de les historiser.
– *Optimisation* : l'ontologie des processus est amenée à évoluer.

## 3.3 Brève revue des formalisations des processus métiers

Il existe différentes méthodes pour effectuer des calculs plus ou moins complexes sur les données, certains sont réalisables dans le langage de requête proposé par le $SGBD$ tandis que d'autres nécessitent des calculs difficiles à exprimer de cette façon. Suivant le cas de figure l'implémentation va être différente. Il sera par exemple possible dans certains cas de mettre en place des vues matérialisées alors que pour les autres il s'avère nécessaire de réaliser des traitements par lots (*batch programs*). Suivant les besoins et le volume de données à traiter ces traitements peuvent être effectués à diverses fréquences.

Toutefois ces différentes solutions relèvent d'un certain niveau de technicité qui ne correspond pas à nos problématiques. Ce constat est la base du *Business Process Management* ou gestion des processus métiers. Cette discipline vise à analyser les processus nécessaires au bon fonctionnement d'une entreprise ou d'un projet pour, dans un premier temps, les modéliser puis, dans un second temps, les automatiser.

Pour la première étape de modélisation il existe plusieurs formalismes :

– *UML*, peut être le plus connu, avec les diagrammes d'activités ou de séquences qui permettent d'établir l'ordre des opérations et les interactions avec les acteurs ou les systèmes.

- *XPDL* (*XML Process Definition Language*), [136] un dérivé d'XML aux processus, mis en place par la Workflow Management Coalition, c'est un standard disposant de ces balises XML.
- La méthode *OSSAD* (Office Support Systems Analysis and Design) ([2]), conforme à la norme ISO 9000:2000 sur les exigences de modélisation, est le fruit du *European Strategic Programme for Research in Information Technology*. C'est une méthode d'analyse d'organisation par les processus.
- *BPMN* (Business Process Modelisation and Notation) est une notation graphique standardisée, maintenue par l'*Object Management Group* [81], dont l'objectif est d'être compréhensible par tous, depuis l'expert d'un domaine au développeur chargé de l'implémentation et l'automatisation du processus.

Ces formalismes sont intéressants et plus particulièrement *BPMN* qui vise la plus grande compréhension possible. Toutefois le passage à la deuxième étape, l'implémentation et l'automatisation est un problème plus complexe. [24] propose une approche sémantique pour d'abord modéliser les processus puis ensuite exploiter les outils sémantiques (raisonneur par exemple) pour en commencer l'exploitation. [53] et [95] combinent des éléments de web sémantique avec l'approche *Business Process Modelisation* pour la «mécaniser» et améliorer les échanges entre les experts et les développeurs.

## 3.4 Méta-modèle de processus avec BPMN

Pour modéliser ces opérations nous avons choisi d'utiliser une modélisation connue et reconnue : *Business Process Modelisation and Notation* (*BPMN*). La figure 5.9 présente une partie des éléments présents dans *BPMN*. On retrouve un élément central générique *BPMN element* qui est spécialisé en quatre grandes familles d'éléments de processus.

- Les *artifacts* permettent de représenter des données, par exemple des entrées ou des sorties.
- Les *swim lanes* organisent les processus, elles permettent, entre autre, de gérer l'échange de données entre des processus et de visualiser leur ordonnancement.
- Les *connecting objects* sont là pour relier des éléments *flow objects* entre eux.
- Les *flow objects* sont au cœur de la modélisation *BPMN*. Ce sont ces éléments qui vont former les différentes étapes d'un processus métier.

Parmi les *flow objects* on va retrouver tous les types d'éléments nécessaires à la réalisation d'un processus :

- Les *events* comme ceux représentés dans la figure 5.9 (*Start, Intermediate et End*) structurent le *PM*, notamment en gérant les conditions de lancement du *PM*.
- Les *activities* s'attachent à décrire dans le détail les actions à réaliser. Ainsi on va retrouver les éléments nécessaires à l'exécution d'une tâche ou d'un sous-processus.
- Enfin les *gateways* autorisent la description explicite des chemins d'exécution d'un processus en fonction du résultats d'*activities*.

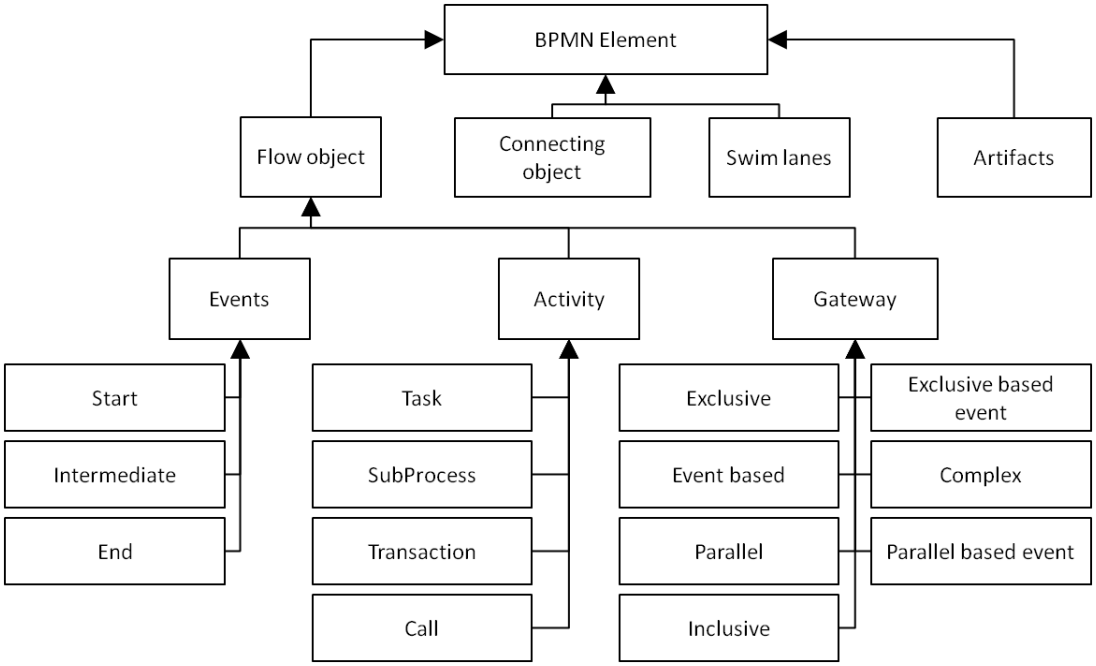


FIGURE 5.9 – Une partie des éléments proposés par BPMN

## 3.5 Implémentation dans *OntoDB* et exemples

### 3.5.1 Implémentation

La création des processus métiers et leur association avec les concepts de l'ontologie des connaissances s'effectue en plusieurs étapes. Il existe plusieurs choix d'implémentation possibles, chacun de ces choix possède un niveau d'abstraction différent et requiert différents niveaux de compétence en terme de manipulation du langage *OntoQL* :

– Pour respecter strictement la définition des *PM* leur implémentation doit s'effectuer entre la partie méta-schéma, dans laquelle on va décrire les différents éléments de BPMN (figure 5.9), et la partie modèle qui va accueillir des instances de ces éléments.
La création des éléments pour le modèle est réalisée par des commandes comme : *ADD ENTITY #BpmnActivity UNDER #FlowObject* pour créer l'élément *Activity* qui hérite de *FlowObject*. A cette entité on va rajouter des propriétés comme un numéro de version : *ALTER ENTITY #BpmnActivity ADD #version STRING*.
En répétant ce processus on exprime les éléments BPMN dans la partie méta-schéma. On ajoute ensuite le méta-modèle d'un *PM*, en indiquant qu'il est composé d'éléments BPMN, voir en précisant des règles, par exemple il doit posséder au moins un élément *Start*.
Ensuite, pour créer les *PM* on va instancier des éléments du méta-modèle dans la partie modèle. Par exemple on va créer le *PM* pour la facturation : *CREATE #BusinessProcess Invoice*, puis les éléments qui le composent : *CREATE #Task GetUserCharge*, et enfin les

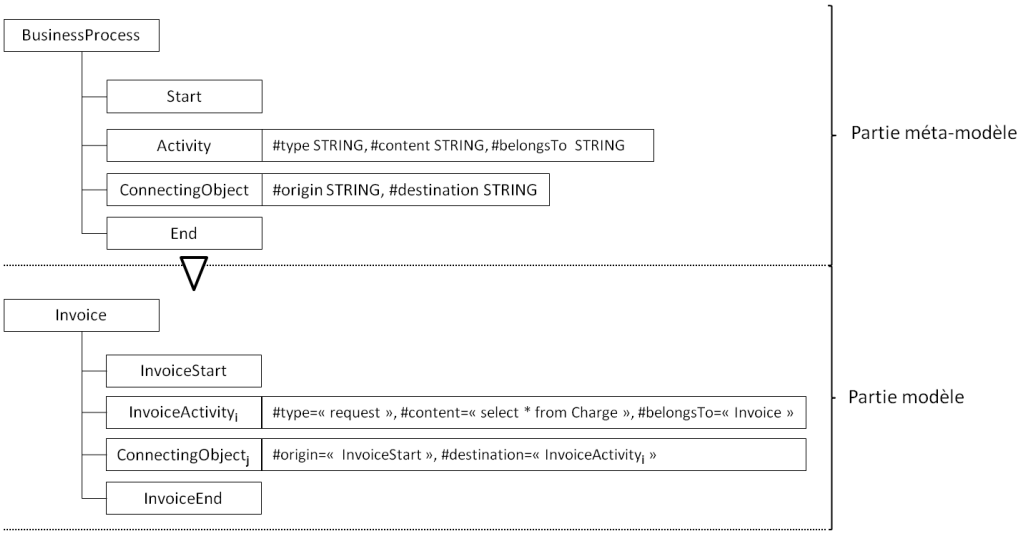


FIGURE 5.10 – Implémentation des PM dans les parties modèle et méta-modèle

connecteurs entre les éléments précédemment définis.
Dans ce cas de figure toute la partie *PM* est cantonnée aux parties modèle et méta-modèle de la plate-forme, la figure 5.10 illustre cette implémentation.

– Une autre façon de procéder consiste à tirer parti de la capacité à stocker des données avec leurs modèles et méta-modèles de la plate-forme *OntoDB*. C'est-à-dire que les détails des éléments BPMN ne seront pas stockés dans la partie modèle mais dans la partie donnée de la plate-forme.
A ce moment là, la partie méta-modèle accueille des descriptions plus génériques des éléments, comme les noms dans différentes langues, un code pour référencer la version du modèle, des descriptions en plusieurs langues, etc.
Pour réaliser cela on exploite l'entité *#Class* déjà présente dans le méta-modèle, cette entité sert à définir des classes ontologiques, on peut l'étendre pour correspondre à nos besoins ou du moins clarifier la différence entre les éléments ontologiques et ceux de BPMN. Ainsi un *PM* va être créé par : *ADD ENTITY #BusinessProcess UNDER #Class*.
Dans la partie modèle on va définir des éléments génériques, contrairement à l'autre implémentation : *CREATE #BusinessProcess EdfBusinessProcess*. On ajoute également les éléments qui nous intéressent *Flow object* et *Connecting object*. Ces éléments sont complétés par des propriétés, comme : *ALTER EdfBusinessProcess ADD name STRING* ou par des références : *ALTER EdfBusinessProcess ADD hasFlowObject ARRAY REF(FlowObject)*.
Ces nouveaux éléments sont étendus dans la partie donnée avec la commande *CREATE EXTENT* déjà abordée dans le chapitre précédent.
Enfin les *PM* et leurs composants sont renseignés dans la partie donnée, le résultat est présenté dans la figure 5.11.

L'implémentation dépend directement des utilisateurs et de leurs besoins. Nos travaux sont destinés à un public d'experts d'un domaine, à travers la création de l'ontologie ces experts

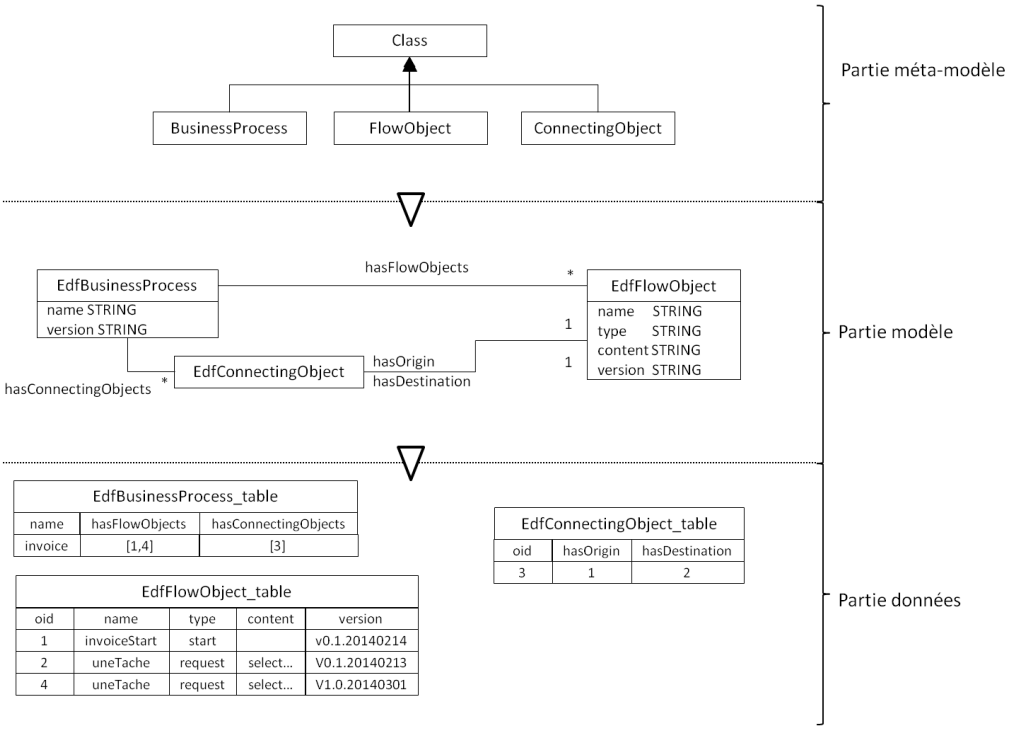


FIGURE 5.11 – Implémentation des PM dans les parties données, modèle et méta-modèle

sont amenés à utiliser et maîtriser des éléments du langage *OntoQL* relatifs à la manipulation de concepts ontologiques et aux données. Or, pour manipuler n'importe quel élément de la partie modèle ce sont les mêmes termes d'*OntoQL* qui sont utilisés. Ainsi en mettant en œuvre une implémentation axée sur la partie modèle et sur la partie donnée les experts peuvent ré-exploiter les compétences précédemment acquises.

On remarque que l'économie de l'apprentissage de la partie relative aux méta-modèles d'*OntoQL* ne dégrade pas les performances de la démarche et permet une mise en place ainsi qu'une prise en main plus rapide par les utilisateurs. Dans notre contexte guidé par les coûts ce facteur est déterminant, c'est pourquoi nous avons opté pour une implémentation sur les trois parties (méta-modèle, modèle, données).

D'autre part, cette implémentation possède un avantage indéniable : elle facilite le versionnage. Ainsi, en plus de disposer d'un historique des connaissances à travers l'entrepôt de connaissances, on dispose de l'historique des *PM* utilisés.

Enfin une telle approche permet de créer naturellement une bibliothèque d'éléments pour concevoir des nouveaux *PM*.

La section suivante présente des exemples de *PM* mis en œuvre dans le cadre de nos travaux.

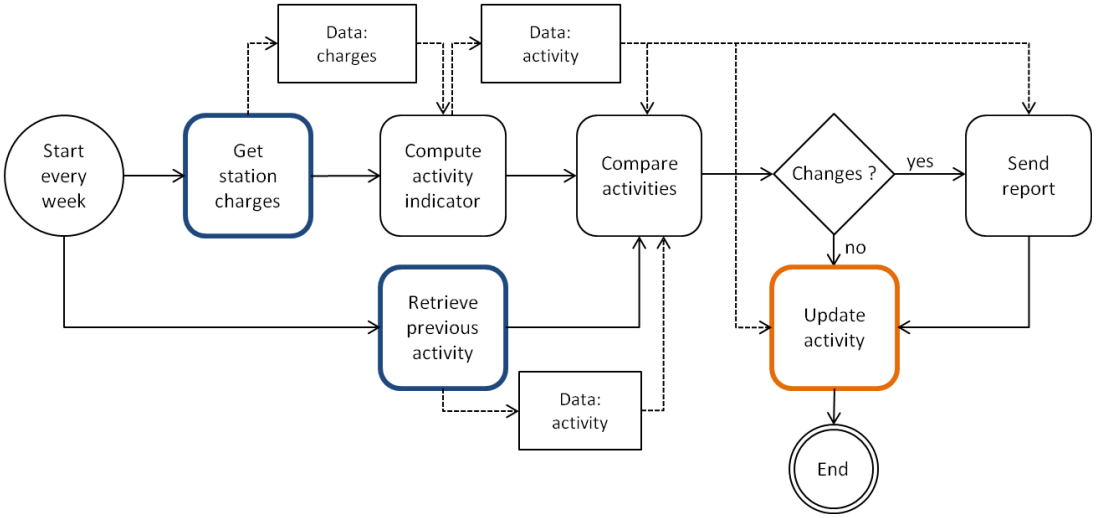


FIGURE 5.12 – Calcul et rapport de l'activité d'une station

### 3.5.2 Exemples de processus métiers

**3.5.2.1 Activité des infrastructures.** La figure 5.12 illustre le *PM* permettant de calculer l'activité d'une station et de reporter tout changement important. Les tâches en bleu sont des requêtes pour interroger les données et la tâche en orange est une requête pour écrire des nouvelles données. Cette analyse requiert deux éléments : les indicateurs des périodes précédentes et ceux de la période en cours. La comparaison de ces indicateurs, associée à une marge de tolérance, permet de détecter des changements ou des anomalies. Dans la suite de ce chapitre figure un exemple d'utilisation de ces connaissances où nous avons pu détecter et corriger des problèmes liés aux horloges de certaines bornes.

**3.5.2.2 Profil utilisateur.** Le *PM* permettant de mettre à jour le profil d'un utilisateur (figure 5.13) ressemble à celui permettant de surveiller l'activité des infrastructures dans la mesure où il fait appel à l'état actuel de l'utilisateur et à l'agrégation de ses états antérieurs.

**3.5.2.3 Groupe d'utilisateurs.** Les groupes d'utilisateurs sont définis à partir des profils actualisés des utilisateurs (figure 5.14), pour un groupe donné il faut : enregistrer les éventuels changements mineurs, signaler les changements importants (*i.e.: les utilisateurs qui apparaissent ou disparaissent du groupe*) et mettre à jour. Ce *PM* ne fait pas appel aux données du domaine, seulement aux connaissances.

**3.5.2.4 Facturation.** La facturation (figure 5.15) peut s'effectuer avant la mise à jour des groupes d'utilisateurs afin que le tarif appliqué à un utilisateur dépende du groupe auquel il appartenait durant la période considérée.

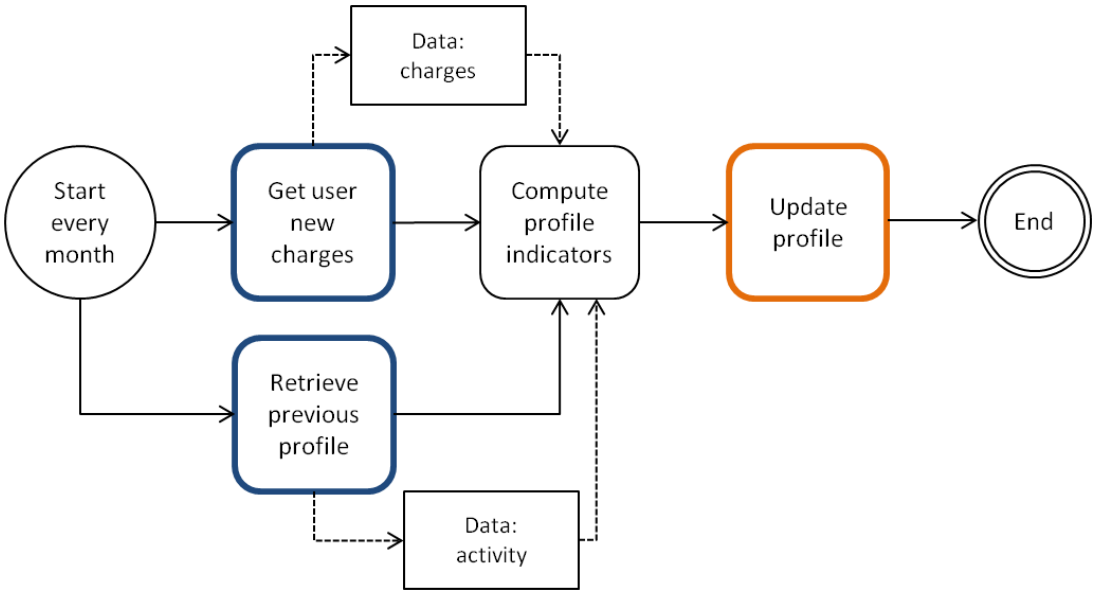


FIGURE 5.13 – Mise à jour du profil d'un utilisateur

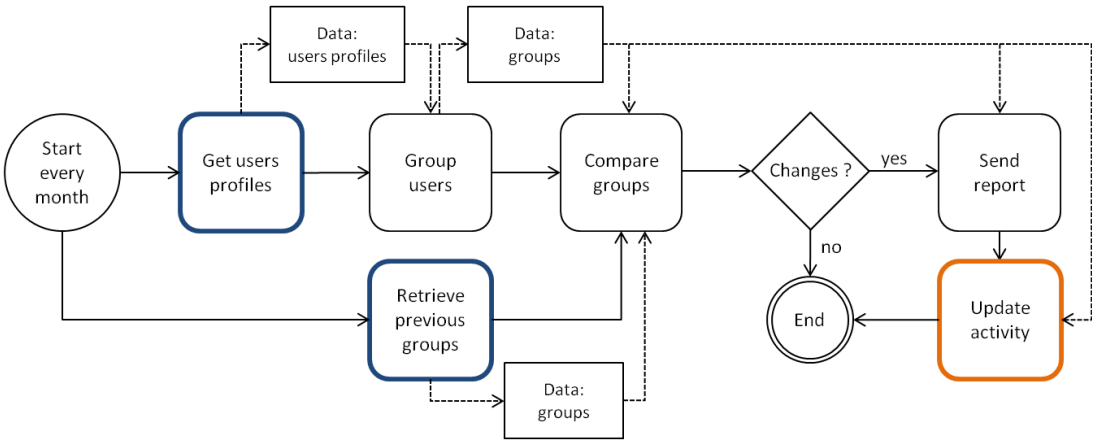


FIGURE 5.14 – Classification des utilisateurs

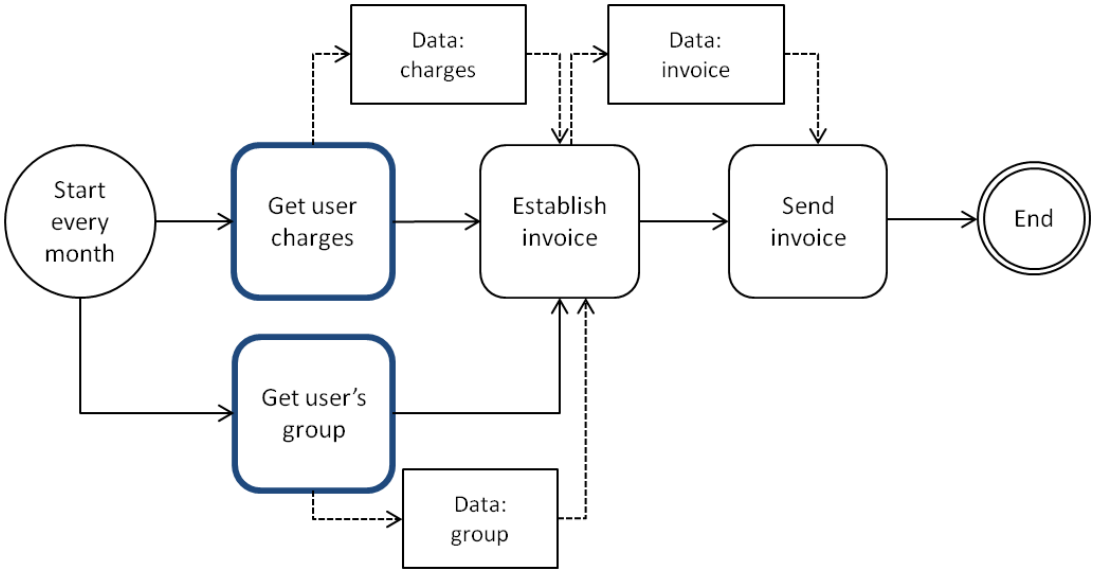


FIGURE 5.15 – Réalisation et envoi d'une facture d'un utilisateur

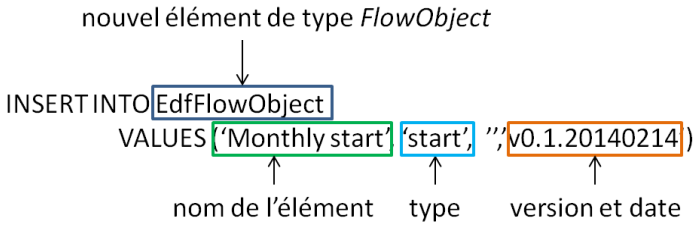


FIGURE 5.16 – Ajout de l'élément Start, ici pour périodicité mensuelle

Pour créer ce *PM* il faut d'abord mettre en place les différents éléments BPMN qui le composent : le début, les tâches, la fin et les connecteurs. Pour illustrer la création du *PM* permettant la facturation, la figure 5.16 et 5.17 montre respectivement la création de l'élément *Start* et de la tâche *GetUserCharges*. La figure 5.18 montre la création du lien entre ces deux éléments (les références des objets à lier sont à lire dans la table des *EdfFlowObject*, cf figure 5.11), l'association de ces éléments au sein d'un *PM* est présentée sur la figure 5.19.

# 4 Entrepôts de connaissances

Si l'on récapitule les éléments à assembler on dispose de : deux ontologies (domaine et connaissances), d'un entrepôt de données et d'un entrepôt de connaissances et des processus métiers. Pour implémenter ces éléments nous avons continué à utiliser la plate-forme *OntoDB*.

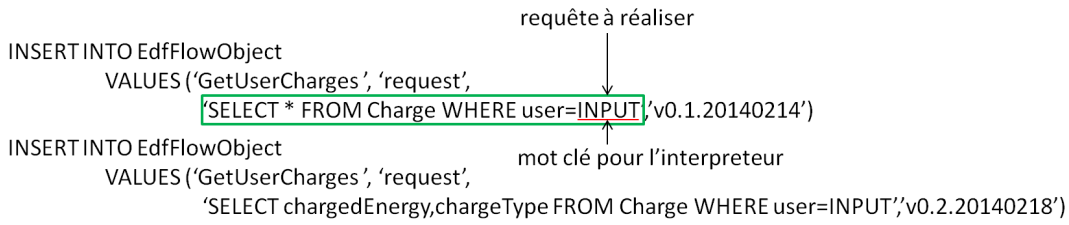


FIGURE 5.17 – Ajout de la tâche chargée de récupérer les charges d'un utilisateur

```
INSERT INTO EdfConnectingObject  VALUES (1,3)
```

FIGURE 5.18 – Création de l'élément permettant de relier les éléments 'Start monthly' et 'GetUserCharges'

## 4.1 Définition et intérêts

Les *ED* sont exploités pour générer des rapports afin d'aider les processus de prise de décisions : choix des articles à acheter par un magasin en fonction des ventes des années précédentes, proposer des nouveaux services de téléphonies à un client en fonction des services utilisés par des clients du même âge, etc. On remarque que les *ED* exploitent les données «brutes» ou plus généralement les données du domaine (les ventes, les services, etc.). Ce fonctionnement peut être comparé aux systèmes fonctionnant en boucle ouverte en automatisme (voir figure 5.20). Dans ce fonctionnement, des décisions sont prises par les décideurs à partir des données et de leur expérience ou de leurs modèles.

Un entrepôt de connaissances (*EC*) possède la même structure qu'un entrepôt de données, il dispose de tables de faits et de dimensions. L'*EC* est caractérisé par les informations stockées dans les tables de faits. En effet, il ne s'agit plus de données du domaine mais des connaissances générées à partir de celles ci. Les connaissances contenues dans l'*EC* sont celles de l'ontologie des connaissances précédemment décrite.

Les *ED* génèrent des rapports à partir des données du domaine pour supporter la prise de décisions, ces rapports sont ensuite exploités par les décideurs grâce à leur expertise du domaine. Le premier intérêt d'un *EC* est de conserver les connaissances contenues dans les rapports.

En disposant des connaissances du passé il devient possible de créer des connaissances plus

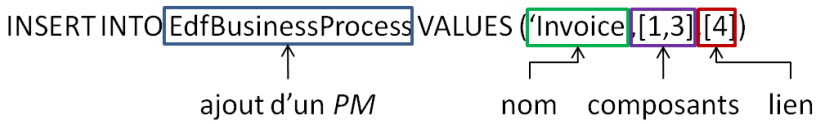


FIGURE 5.19 – Création du PM r'Invoice' utilisant les éléments précédemment décrits

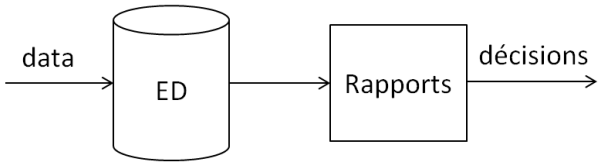


FIGURE 5.20 – Analogie de l'utilisation d'un ED avec les boucles ouvertes en automatique

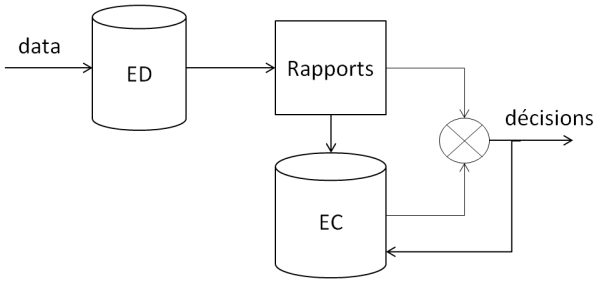


FIGURE 5.21 – Analogie de l'utilisation d'un ED et d'un *EC* avec les boucles fermées en automatique

pointues et de mesurer les évolutions dans le temps. On peut alors mesurer la variation d'un état à un autre plutôt que de se baser sur le dernier état connu pour réaliser des analyses. De plus, en conservant une trace des décisions prises, représentées comme une connaissance, on peut alors mesurer leur impact sur le domaine. Cette aspect est particulièrement important, notamment pour le volet commercial des décisions.

L'expérience et les modèles sont bâtis sur l'analyse «manuelle» des résultats des décisions. Nous avons choisi d'intégrer ce processus de *feedback* à notre plate-forme en formant un entrepôt de connaissances flottant défini dans le paragraphe suivant. Pour reprendre l'analogie avec l'automatisme, cela permet de passer un système en boucle fermée (figure 5.21).

## 4.2 Entrepôt de connaissances flottant

On dispose à présent d'un *EC* capable d'accueillir des connaissances, décrites dans une ontologie, ainsi que des *PM* permettant de générer ces connaissances. La connexion entre ces éléments permet de mettre en place un entrepôt de connaissances flottant.

### 4.2.1 Définition

L'*ED* traite et enregistre des données issues du domaine, ces données vont servir à générer des connaissances. Lorsqu'elles peuvent être générées, les connaissances sont stockées dans l'*EC*. On parle alors d'entrepôt de connaissances flottant (*ECF*) car le peuplement de l'*EC* est dépendant des connaissances pouvant être générées, et donc des données disponibles.

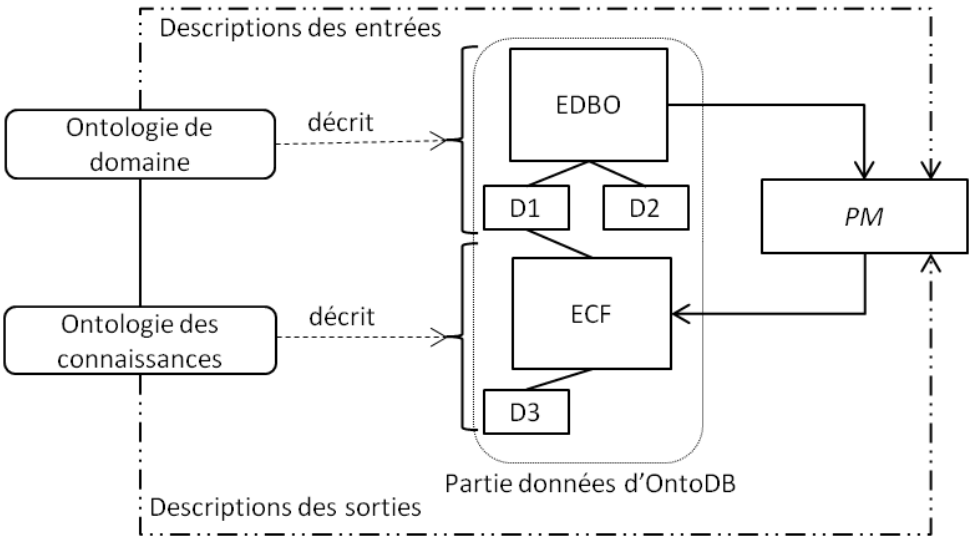


FIGURE 5.22 – Fonctionnement de l'ECF (D1, D2 et D3 représentent des dimensions partagées par les deux entrepôts)

### 4.2.2 Déploiement

La première étape consiste à créer l'*ECF* en commençant par étendre l'ontologie des connaissances dans la partie donnée d'*OntoDB*. Cette démarche s'effectue comme pour la création de l'*EDBO*. Pour peupler l'*ECF*, il faut s'assurer qu'à la fin de leurs exécutions les *PM* enregistrent les connaissances qu'ils génèrent.

Il faut ensuite développer un programme qui va se charger de vérifier si des *PM* doivent être lancés. Pour cela des mots clés ont été choisis pour nommer les éléments *start* des *PM*. Pour reprendre l'exemple de la facturation, les *PM* démarrant par *start monthly* seront exécutés le premier de chaque mois. La figure 5.22 illustre le mode d'alimentation de l'*ECF*, D1, D2 et D3 représente des dimensions, communes ou non à l'*EDBO* et à l'*ECF*.

### 4.2.3 Évolution des travaux des experts avec cet outil

Jusqu'à présent, la génération des connaissances devait être réalisée par chaque expert souhaitant les manipuler. On était donc en présence de tâches répétitives et redondantes. Le temps que passait un expert à calculer ces connaissances représentait du temps en moins pour les exploiter et leur ajouter de la valeur par son expertise.

Avec l'*ECF* opérationnel, c'est-à-dire un entrepôt dans lequel les connaissances sur le domaine sont générées automatiquement, les experts peuvent accéder aux bases de connaissances par des requêtes simples, du type *select * from Invoice*. Les experts peuvent donc se consacrer uniquement au traitement des connaissances, et non plus aux techniques de génération des connaissances.

Il y a donc moins de travail technique (sans valeur ajoutée), et d'avantage d'expertise. A ce stade nous disposons d'une plate-forme qui intègre tout le cycle de vie des données : depuis leur intégration et leur description jusqu'à leur utilisation et leur restitution par la plate-forme. Le tout étant complètement guidé par les éléments déclaratifs que sont les ontologies (de domaine et des connaissances) et des *PM*.

# 5 Conclusion

La formalisation des connaissances en une ontologie offre à l'entreprise un système d'informations pour capitaliser et ré-utiliser des connaissances. Elle représente également un gain de temps en terme de recherche de connaissances, d'experts ou de méthodes. L'extension de la description des connaissances en processus métiers exécutable ajoute une nouvelle dimension en terme d'exploitation.

En effet, grâce aux *PM* qui leurs sont associés, les connaissances sont automatiquement générées. Ainsi la partie connaissance de l'ontologie, *i.e.* les connaissances et les *PM*, ne forme plus juste un catalogue mais aussi une prolongation du cycle de vie de la donnée. Cette prolongation correspond à la génération de la connaissance mais aussi à la gestion des moyens de la calculer qui deviennent des éléments historisés et traçables. Le temps investi dans la construction des *PM* est donc récupéré rapidement par l'automatisation. Cela permet aux experts de se focaliser sur leur spécialité et l'interprétation des résultats plutôt que sur des calculs répétitifs. Cette démarche est illustrée à travers les exemples d'analyses exposés dans ce chapitre.

De plus, le traitement de la connaissance au sein d'un entrepôt de données, re-baptisé entrepôt de connaissances, permet de construire des nouvelles connaissances, plus précises, se basant sur les données du domaine et sur les connaissances déjà présentes. Cet entrepôt de connaissances peut être qualifié d'entrepôt flottant dans la mesure où son alimentation dépend de l'*EDBO* de domaine.

Le chapitre suivant détaille des résultats obtenus grâce à l'application de notre solution complète.

# Troisième partie

# Cas d'étude et conclusions

Chapitre 6

# Cas d'étude EDF

## Sommaire

# 1 Introduction

Ce chapitre est consacré aux résultats obtenus par la mise en place de notre solution pour EDF dans le cadre du projet «Véhicules Électriques».

La mise en place d'abord de l'ontologie de domaine puis de celle des connaissances ont permis de formaliser le domaine et de capitaliser les différents travaux accomplis par les experts du projet «Véhicules Électriques». En ajoutant les processus métiers aux concepts de l'ontologie des connaissances les analyses déjà menées ont pu être actualisées et réalisées sur différents corpus de données. Ces corpus correspondent aux données recueillies dans différentes expérimentations, elles sont présentées dans la suite de ce chapitre. Une partie de ces résultats est présentée dans ce chapitre.

Après un bref historique permettant de situer l'évolution du $\mathcal{VE}$ par rapport aux travaux en cours et à venir, nous présentons les expérimentations dont sont issues les données traitées dans les sections suivantes. Ces dernières montrent une partie des analyses menées grâce à notre solution. Ce chapitre se conclut sur la compilation de nombreuses connaissances pour mettre au point un modèle d'affaire basé sur la théorie des jeux.

# 2 Bref historique du véhicule électrique

## 2.1 1890-1930

L'exploitation des $\mathcal{VE}$ a débuté au début du XX$^{e}$ siècle : utilisation individuelle, compagnies de taxis, courses de vitesse, etc. Elle s'est poursuivie pendant quelques décennies puis l'évolution des moteurs à énergie fossile a progressivement fait disparaître les $\mathcal{VE}$. On peut citer différentes causes de ce déclin comme le développement des réseaux routiers entre les villes qui a fait apparaître le besoin d'une autonomie plus élevée, de l'invention du starter électrique qui a permis aux véhicules thermiques de se passer d'un démarreur à manivelle ou encore du développement aux États-Unis des forages pétroliers qui a fait chuter le cours du carburant.

## 2.2 1930-1990

Les chocs pétroliers ainsi que l'embargo pétrolier contre les États-Unis ont relancé l'intérêt des constructeurs et des usagers pour le $\mathcal{VE}$. Cette relance a été soutenue par des lois comme la *Electric and Hybrid Vehicle Research, Development, and Demonstration Act* aux États-Unis en 1976. Toutefois les projets n'ont pas rencontré le succès escompté. Notamment à cause de la faible autonomie des véhicules (autour de 100 km), ou des vitesses trop faibles (la Vanguard-Sebring pouvait monter seulement à 48 km/h).

## 2.3 1990-aujourd'hui

Il faudra attendre le début des années 90 pour que de nouveaux projets voient le jour. Outre le prix du carburant, toujours en hausse, s'est ajouté l'intérêt pour des véhicules plus propres générant moins de pollution ainsi que des progrès importants sur les technologies de batteries. Cet intérêt a été entériné par des textes de loi comme le *Zero Emission Vehicle* en Californie qui impose aux constructeurs de vendre au moins 2% de véhicules verts. Ainsi la combinaison de l'économie, de l'écologie et de l'amélioration des technologies a permis au $\mathcal{VE}$ de prendre un nouvel essor.

En France cet essor est également soutenu par les pouvoirs publics : mise en place de prime à l'achat, participation de l'Agence De l'Environnement et de la Maîtrise de l'Énergie (ADEME), création de crédits d'impôts [109], etc. Ce soutien a été inscrit dans la loi [49] dans le cadre de la transition énergétique. Cette loi prévoit la mise en place d'un réseau de 17 millions de bornes de recharge à l'horizon 2030, le remplacement d'un véhicule d'état sur deux par un $\mathcal{VE}$ ou un $\mathcal{VHR}$, l'obligation pour les entreprises d'avoir des $\mathcal{VE}$ dans leurs flottes de véhicules, etc. La France a également mis en place le Programme de Recherche Et D'Innovation dans les Transports Terrestres (PREDIT) [32], il s'agit d'un «*dispositif interministériel dédié à la mise en cohérence des actions publiques de mobilisation de la recherche et de l'innovation dans les transports terrestres*». Le PREDIT en est à sa cinquième édition, la quatrième édition avait financé un millier projets pour un montant d'1,2 milliard d'euros.

## 2.4 Perspectives d'évolution

L'essor depuis les années 90 est plus soutenu en raison de plusieurs facteurs : la hausse significative des prix des différents carburants et la volonté politique de diminuer la pollution locale des véhicules thermiques. Ces nouvelles contraintes d'ordre économique et écologique et l'évolution des technologies ont favorisé ce nouvel essor du $\mathcal{VE}$

Tout d'abord l'évolution des technologies de batteries a permis d'augmenter l'autonomie des $\mathcal{VE}$. Ensuite, un $\mathcal{VE}$ n'est pas utilisé comme un véhicule thermique : les trajets réalisés sont majoritairement des trajets *quotidiens* et *connus*. Toutefois le risque de trajets, ou de conditions de circulation, imprévus restent de véritables freins à l'achat. Cependant l'augmentation de l'autonomie des batteries, grâce à des technologies permettant des concentrations importantes d'énergie, a permis d'élargir le rayon d'action et donc l'éventail d'usage du $\mathcal{VE}$.

Pour soutenir ces considérations économiques et écologiques nous disposons aujourd'hui de la technologie nécessaire pour quantifier les déplacements et communiquer ces informations (vitesses au cours du trajet, état du véhicule, destination, etc.). Cela grâce aux téléphones portables ou par des véhicules très informatisés, comme la Tesla [131]. Ces informations permettent aux acteurs d'améliorer leurs offres par rapport aux besoins de mobilité des utilisateurs et donc de favoriser l'utilisation des $\mathcal{VE}$.

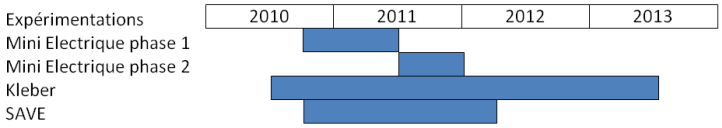


FIGURE 6.1 – Diagramme de Gantt des expérimentations dont les données ont été exploitées dans cette thèse

# 3 Expérimentations

Les expérimentations menées ces dernières années sont le signe d'une nouvelle approche : auparavant les $\mathcal{VE}$ étaient vendus comme des véhicules thermiques, cette vision a changé. Afin de montrer que le $\mathcal{VE}$ a sa place dans la société (différents usages, multi-utilisateurs, avec des services connectés, etc.) des expérimentations fondées sur une véritable interaction entre les collectivités locales, les constructeurs, les fournisseurs de services, les utilisateurs, etc. se sont mises en place. Ces expérimentations mettent ces aspects en valeur, et dans le même temps elles servent à connaître d'avantage la mobilité électrique ($ME$).

La majorité de ces expérimentations est dirigée par les constructeurs de véhicules électriques (BMW [22], Toyota, Renault, etc.). Toutefois certaines expérimentations sont menées par des entreprises (La Poste par exemple [103], ou EDF avec sa propre flotte). Le groupe EDF est impliqué, et a été impliqué, dans plusieurs expérimentations ces dernières années. Son rôle a été d'intervenir dans la fourniture d'électricité ainsi que dans l'analyse des données récupérées lors des charges des $\mathcal{VE}$. Voici les différentes expérimentations suivies par EDF (voir figure 6.1).

## 3.1 BMW et EDF : Mini Électrique

Cette expérimentation mise en place par BMW s'est déroulée en deux phases, une première de 6 mois entre décembre 2010 et juin 2011 et la seconde de 5 mois entre aout 2011 et décembre 2011. Les deux phases se sont déroulées en région parisienne. La flotte de Minis Électriques était composée d'une cinquantaine de véhicules, répartis pour moitié entre des particuliers et des entreprises partenaires. La première phase a permis de tester et d'améliorer des boitiers de charge ce qui a permis, par la suite, d'améliorer le recueil de données pour la deuxième phase.

La mini électrique utilisée est un prototype qui n'a pas été produit en série. Elle est dotée d'un moteur de 150 kW, soit environ 200 Ch, pour une autonomie donnée de 240 km.

Les conditions expérimentales étaient particulières dans la mesure où l'on avait un lien exclusif entre une borne, au domicile ou sur le parking de l'entreprise, et un véhicule, il n'y avait pas de notion de réseau de charge.

22. Bayerische Motoren Werke : constructeur automobile allemand

## 3.2 Toyota, l'école des Mines de Paris et EDF : Kleber

Toyota a réalisé une expérimentation sur 3 ans, entre 2010 et 2013, à Strasbourg avec des Prius hybrides rechargeables, on parle dans ce cas de Véhicule Hybride Rechargeable ($\mathcal{VHR}$). La flotte, une soixantaine de véhicules, a été stable sur les trois ans de l'expérimentation : même nombre de véhicules, même modèle de véhicule et même groupe d'utilisateurs (des professionnels et des utilisateurs de véhicules de fonction).

Ce modèle d'hybride pouvait développer un maximum de 136 Ch (combinaison du moteur thermique et du moteur électrique) et l'autonomie purement électrique se situait autour de 20 km.

## 3.3 Renault, Schneider, EDF : SAVE

Renault est à l'origine du projet Seine Aval Véhicule Électrique (SAVE) dont l'objectif était de tester le déploiement de $\mathcal{VE}$ et de bornes de recharge sur un territoire. L'expérimentation a regroupé environ 80 véhicules sur une période d'un an et demi, entre 2010 et 2012. Renault a pu tester la Kangoo électrique ainsi que différentes versions de la Fluence électrique.

La Kangoo dispose d'un moteur de 60 Ch pour une autonomie de 170 km. La Fluence quant à elle dispose de 93 Ch pour une autonomie de 185 km.

## 3.4 CROME

L'expérimentation transfrontalière, entre la France et l'Allemagne, *CROss-border Mobility for EVs* (CROME) possédait des objectifs différents des autres expérimentations. Parmi les objectifs attendus il y avait un volet important sur la capacité à se charger d'un côté ou de l'autre de la frontière, c'est-à-dire : l'itinérance des services (identification et paiement électronique) et la compatibilité des infrastructures avec les véhicules. Une centaine de $\mathcal{VE}$ de neuf types différents ont été impliqués et une quarantaine de stations de recharge. Les enjeux de cette expérimentation ont fait l'objet d'une publication [39].

# 4 Résultats

Pour illustrer la démarche voici quelques résultats obtenus à partir de la plate-forme dotée de l'*EDBO* et le l'*ED* des connaissances.

## 4.1 Exemples relatifs à la supervision

### 4.1.1 Volume de données

La plate-forme gère environ 43 000 lignes de données, cela représente le nombre de charges enregistrées pendant les expérimentations par un petit nombre de $\mathcal{VE}$, et ces données ont été recueillies par quelques centaines d'équipements différents.

D'après les prévisions du gouvernement le nombre de $\mathcal{VE}$ pourrait atteindre quelques millions dans les prochaines décennies. En travaillant avec le nombre moyen de charges par jour et par véhicule, on obtient deux fréquences. D'abord une fréquence brute qui donne une charge tous les quatre jours pour un $\mathcal{VE}$, soit 90 charges par an. Une fréquence un peu plus juste peut être calculée grâce à l'observation de la répartition des charges sur la semaine, en effet il y a peu de charges le week-end : besoins de mobilité moins prévisibles, trajets plus longs, etc. En considérant les jours ouvrés plutôt que tous les jours de la semaine, la fréquence (moyenne) obtenue est d'une charge tous les trois jours, soit 124 charges par an. A partir de cette fourchette de fréquences il devient possible d'extrapoler le volume de données à $10^6$ $\mathcal{VE}$. Cela donne entre 90 et 124 millions de charges à enregistrer par an, si l'on suppose qu'une charge est représentée par 12 octets (un octet par information en moyenne) alors on atteint des volumes de l'ordre du giga-octet par an. Ce volume extrapolé est une grossière approximation mais il permet de montrer que l'ordre de grandeur reste dans les capacités des plates-formes déjà existantes (capable de gérer des téra-octets de données). Le stockage des connaissances en plus des données ne devrait pas, si l'on se base sur les connaissances actuellement gérées, dépasser le volume de données du domaine et donc le nombre de données ne devrait pas être multiplié par un facteur supérieur à 10. Cette extrapolation est à manipuler avec précaution et doit être exprimer dans son contexte. Le volume de données est amené à s'étoffer avec le développement de nouveaux services.

### 4.1.2 Détection de pannes

Comme cela apparaît dans l'ontologie les bornes appartiennent à des stations, les stations peuvent posséder un serveur d'acquisition et de contrôle (*SCADA*, cf figure 6.2) ou le partager avec d'autres stations. Cette supervision permet de détecter des pannes à différents niveaux :

- Les pertes de communication entre le *SCADA* et la plate-forme.
- La perte de communication entre une borne et le *SCADA* qui la supervise.
- Des problèmes d'acquisition par les bornes.

La figure 6.3 illustre ce dernier type de panne. On observe que les charges enregistrées au début de l'expérimentation ont une durée nulle, une fois décelé ce problème a été rapidement corrigé par une mise à jour des logiciels dans les bornes. Pendant l'été 2011 ce problème est réapparu, immédiatement détecté il a pu être corrigé.

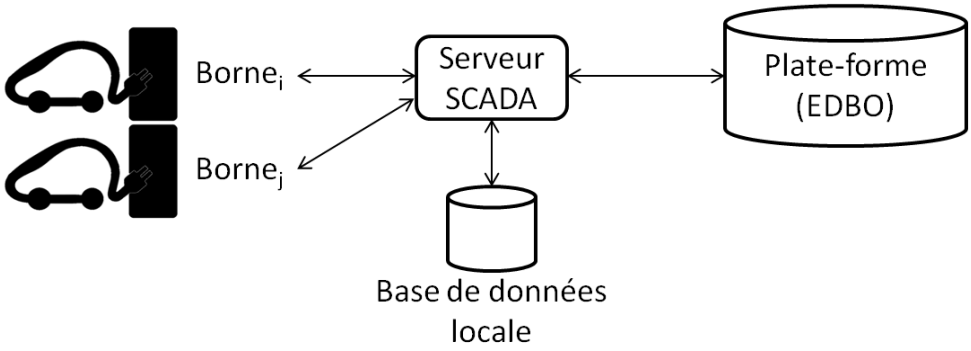


FIGURE 6.2 – Chaîne d'acquisition depuis les bornes jusqu'à la plate-forme d'exploitation pour les expérimentations

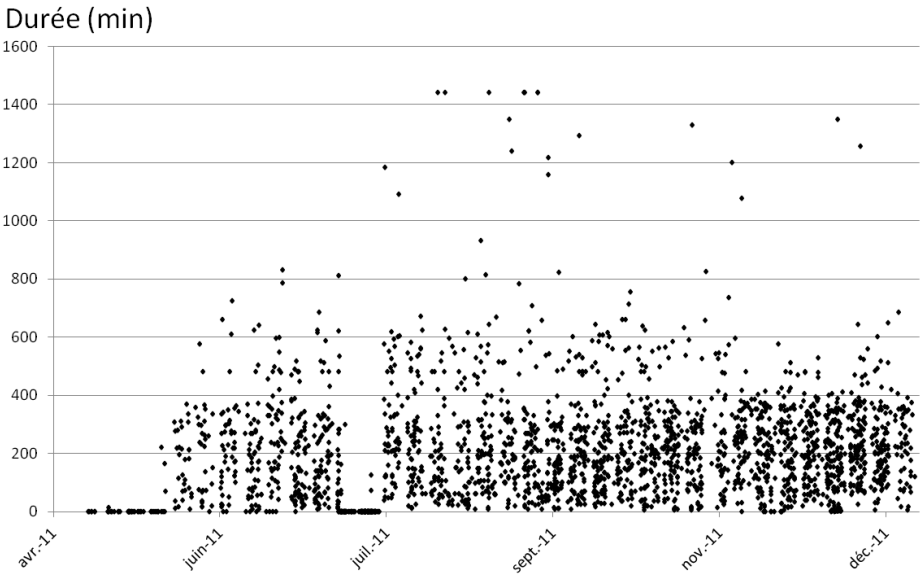


FIGURE 6.3 – Supervision : chaque point représente une charge, l'ordonnée correspond à la durée et l'abscisse à la date (données SAVE)

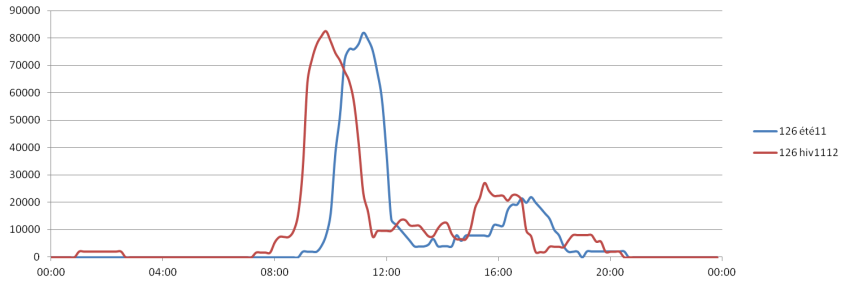


FIGURE 6.4 – Courbe de charges journalière (Puissance en fonction de l'heure) de la borne 126

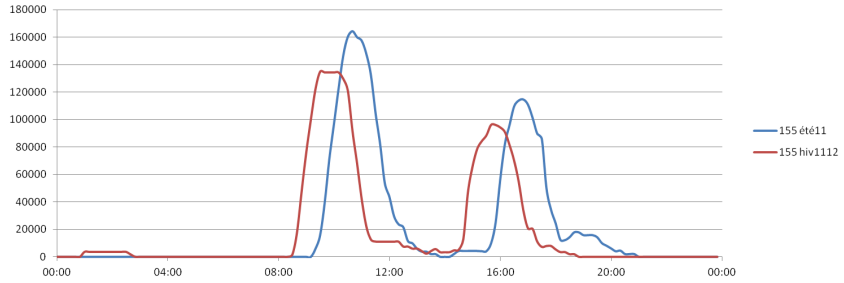


FIGURE 6.5 – Courbe de charges journalière (Puissance en fonction de l'heure) de la borne 155

Cette supervision est simple à mettre en place et à automatiser, et elle permet d'intervenir rapidement sur un certain nombre de problèmes et dans de brefs délais.

### 4.1.3 Détection d'anomalies sur les infrastructures et les charges

**4.1.3.1 Horodatage** Afin d'illustrer la mise en place de la surveillance des infrastructures sous la forme d'un *ED* voici un exemple où cette surveillance a permis de détecter une source d'erreurs autrement à peine visible. La plupart des analyses menées sur les comportements des utilisateurs se basent sur des données agrégées, ces données proviennent de différentes infrastructures et peuvent correspondre à des périodes longues (plusieurs mois). Ce type d'analyses permet d'obtenir des informations pointues mais elles peuvent également masquer des erreurs. La mise en place de la surveillance des infrastructures a permis de relever des changements étonnants : deux fois dans l'année des utilisateurs décalaient leurs habitudes d'une heure.

Sur certaines bornes, on observait exactement le même comportement mais décalé d'une heure, ce changement intervenait du jour au lendemain : les derniers dimanches des mois d'octobre et de mars : les figures 6.5 et 6.4 montre le décalage de l'activité en fonction de la référence temporelle et les figures 6.6 et 6.7 montre l'enregistrement des informations en heure locale (le changement d'heure est pris en compte par l'horloge interne de la borne). Il s'est avéré que les horloges des bornes de recharge ne fonctionnaient pas toutes de la même façon (changement d'heure automatique ou non), ni n'avaient été calibrées de la même façon (mise en place en heure d'été ou en heure d'hiver).

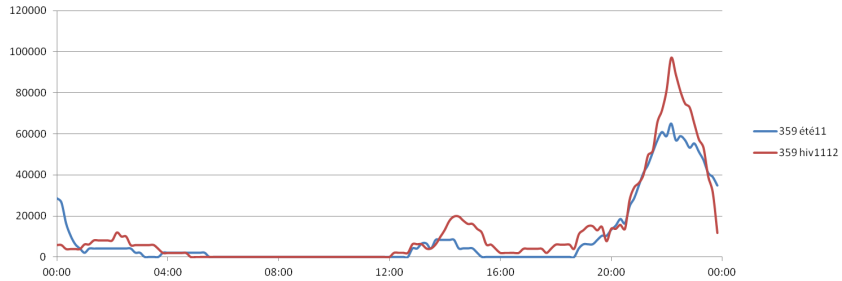


FIGURE 6.6 – Courbe de charges journalière (Puissance en fonction de l'heure) de la borne 359

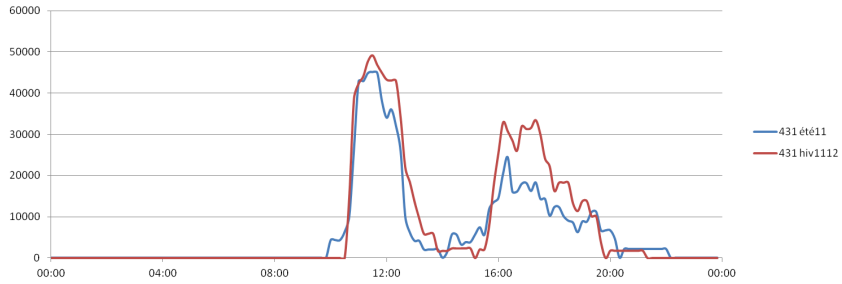


FIGURE 6.7 – Courbe de charges journalière (Puissance en fonction de l'heure) de la borne 431

Ce type d'anomalies peut biaiser les études sur la consommation des $\mathcal{VE}$ pendant les pics de consommation. On observe une diversité croissante parmi les infrastructures de recharge, et il n'est pas possible d'effectuer des contrôles sur toutes les nouvelles installations. En revanche, maintenant que cette faille (heure d'été, heure d'hiver) est identifiée et documentée, il devient possible d'intégrer un processus de correction des données dans la plate-forme. Ainsi, à défaut de pouvoir contrôler les infrastructures au moment de leur mise en place, on peut réaliser une boucle de contrôle.

De plus, maintenant que les outils d'analyse ont été mis en place il n'est plus nécessaire de revenir sur ces problèmes. L'analyse «manuelle» n'est réalisée qu'une seule fois puis elle est formalisée et automatisée dans la plate-forme.

**4.1.3.2 Détection de charges anormales** La description d'une charge dans l'ontologie a permis d'exprimer des contraintes simples telles que : la durée doit être positive ou nulle tout comme l'énergie chargée, ces deux informations doivent être des nombres, etc. Les charges présentant des éléments aberrants sont signalées, cependant, parmi les charges qui sont intégrées certaines présentent des anomalies plus fines à détecter. C'est pourquoi des tris sont décrits dans la partie connaissance pour corriger ces données *via* des processus métiers.

Par exemple, l'un des indicateurs utilisés pour détecter de telles charges est la *monotone des durées*. C'est une courbe où chaque point représente une charge avec comme ordonnée l'énergie chargée et où les charges sont classées par énergie croissante (l'abscisse correspond

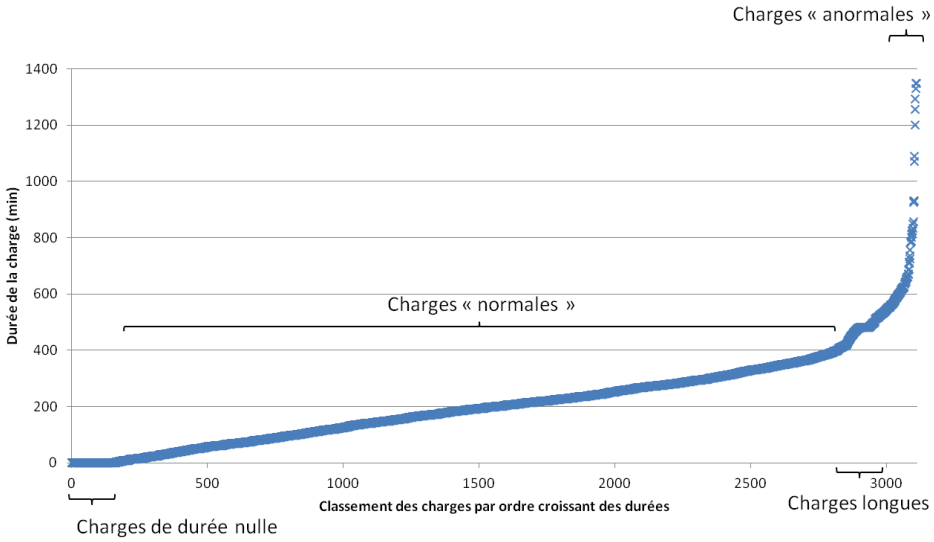


FIGURE 6.8 – Monotone des durées (données SAVE)

au classement de la charge). La figure 6.8 correspond à la monotone des durées pour une partie des charges d'une expérimentation, on observe plusieurs éléments remarquables :

- Au début on retrouve les charges de durée nulle, ces charges peuvent être facilement détectées lors de l'intégration, cet indicateur permet également de les mettre en avant.
- Au centre, sur la partie droite on retrouve environ 80% des données qui représente les charges «normales».
- On remarque ensuite des charges de longue durée qui ne sont pas dans l'alignement des précédentes et dont une partie présente même un plateau à 480 min (8h). Ces charges sont normales et le plateau à 8h a été identifié comme un dispositif de sécurité intégré à un certain type de véhicule pour arrêter la charge au bout de 8h. Accessoirement cela permet d'identifier ces véhicules automatiquement.
- La partie la plus à droite correspond aux charges de très longue durée. Ces charges peuvent être détectées automatiquement car elles sortent de la «normale», elles requièrent l'analyse d'un expert afin de créer des filtres, et/ou des corrections, avec un processus métier.

## 4.2 Exemples de résultats d'études comportementales

Les utilisateurs des $\mathcal{VE}$ sont identifiés par leur badge (dans la majorité des cas), cela permet, dans le cas de véhicules partagés, de distinguer les utilisateurs d'un même $\mathcal{VE}$. Toutefois pour mener certaines études, notamment sur les véhicules partagés, c'est l'identifiant du véhicule qui est exploité.

### 4.2.1 Définition des indicateurs sur les utilisateurs

Afin de comprendre le fonctionnement de la $ME$ plusieurs indicateurs ont été mis en place et testés sur les données issues des expérimentations décrites ci-dessus. Une partie de ces indicateurs sont présentés dans les paragraphes ci-dessous.

**4.2.1.1 Indicateurs globaux** Les premiers indicateurs mis en place sont les indicateurs globaux. Ces indicateurs sont des sommes ou des moyennes qui visent à exprimer la fréquence des charges, l'énergie chargée, etc. Bien que simples à calculer ces indicateurs sont difficilement interprétables, par exemple savoir que la fréquence des charges d'un utilisateur est de 3,7 jours en moyenne n'apporte aucune information exploitable sur son comportement car il faudrait distinguer les charges courtes ou longues, le moment où elles arrivent au cours de la semaine et de la journée, etc. C'est pourquoi il est apparu nécessaire de disposer d'indicateurs plus complexes qui se basent sur l'historique de l'utilisateur et qui combinent les informations des charges afin de disposer d'éléments comparables.

Le premier de ces indicateurs est la courbe de charges présentée dans le paragraphe ci-dessous.

**4.2.1.2 Courbe de charges** A partir de l'identifiant choisi il devient possible de faire des recherches précises sur l'activité d'un utilisateur ou d'un groupe d'utilisateurs et, principalement, de relever les charges effectuées. Ces charges vont permettre de dresser un profil énergétique et temporel des utilisateurs sous la forme d'une courbe de charges ou d'un ensemble de courbes de charges, la définition et le calcul d'une courbe de charges étaient donnés dans le chapitre précédent.

Par exemple la figure 6.9 correspond au profil typique d'un particulier, les charges ont principalement lieu la nuit. En revanche la figure 6.10 montre un profil classique des véhicules d'entreprise mis en charge l'après-midi.

**4.2.1.3 Décomposition en séries de Fourier** Il est possible de créer des courbes de charges sur différentes périodes, dans la plupart des cas ce sont les courbes de charges journalières qui sont construites. Afin d'évaluer la fréquence hebdomadaire des charges les courbes de charges mensuelles ont été produites car elles permettent d'afficher plusieurs semaines. Ensuite, ces courbes mensuelles sont supposées périodiques avec comme période une semaine, cette hypothèse permet de réaliser des décompositions en séries de Fourier [62] afin de décomposer la courbe mensuelle en signaux de différentes fréquences. Puis une lecture des fréquences et des amplitudes des signaux permet de détecter les principales composantes de la courbe de charges mensuelles.

La figure 6.11 illustre cet indicateur, sur cette figure les coefficients $a_n$ et $b_n$ ainsi que $M_Q$ la moyenne quadratique de ces coefficients sont présentés en fonction de la période. Ces courbes

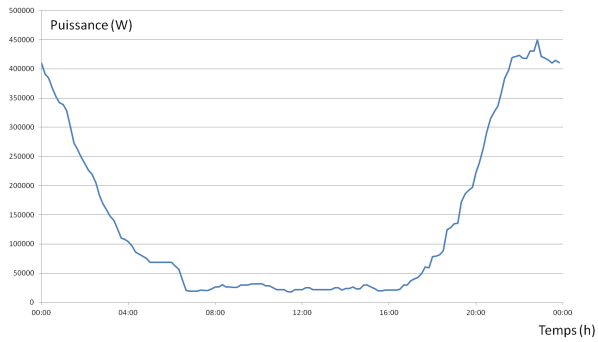


FIGURE 6.9 – Courbe de charges journalière moyenne (Puissance en fonction de l'heure) d'un particulier

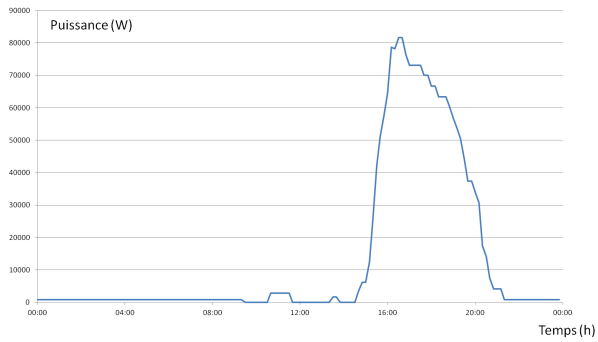


FIGURE 6.10 – Courbe de charges journalière moyenne (Puissance en fonction de l'heure) d'un véhicule de service (plusieurs utilisateurs)

ont été générées en transformant une courbe de charges mensuelle de tous les utilisateurs en série de Fourier, elles présentent quatre informations remarquables :

- Un évènement journalier, c'est-à-dire qui se répète tout les jours, qui correspond à des habitudes journalières : connexion et déconnexion. L'étude détaillée des courbes de charges montre effectivement que certaines habitudes sont régulières à une dizaine de minutes près.
- L'évènement bi-journalier correspond à des connexions et déconnexion réalisées par plusieurs catégories d'utilisateurs, en se superposant cela crée un évènement bi-journalier.
- L'évènement de période 3,5 jours va correspondre à l'intervalle entre les charges avant un week-end et après. Dans la mesure où un certain nombre de participants sont des professionnels, la dernière charge de la semaine à lieu le vendredi matin (pendant la nuit) et la charge suivante intervient le lundi dans la journée.
- Le dernier point concerne le pic pour la période d'une semaine, à l'extrémité du graphique, il s'agit d'un effet lié à la fenêtre d'observation qui est d'une semaine. Toutefois, l'analyse sur une fenêtre plus grande montre une périodicité marquée sur une semaine.

#### 4.2.1.4 Histogrammes durée/fréquence

Une courbe de charges représente une agrégation de données intéressante, toutefois, pour obtenir des précisions sur certains comportements il peut s'avérer nécessaire de disposer de données moins agrégées. Les histogrammes de durée de charge en fonction de leurs fréquences apportent un complément d'information aux analyses de Fourier. Une branche de l'histogramme correspond au nombre de charges dont la durée est comprise dans un certain intervalle.

La figure 6.12 montre que cet indicateur permet d'observer clairement la répartition des durées ainsi que les comportements hors normes, comme l'utilisateur dont la moitié des charges durent 70 minutes.

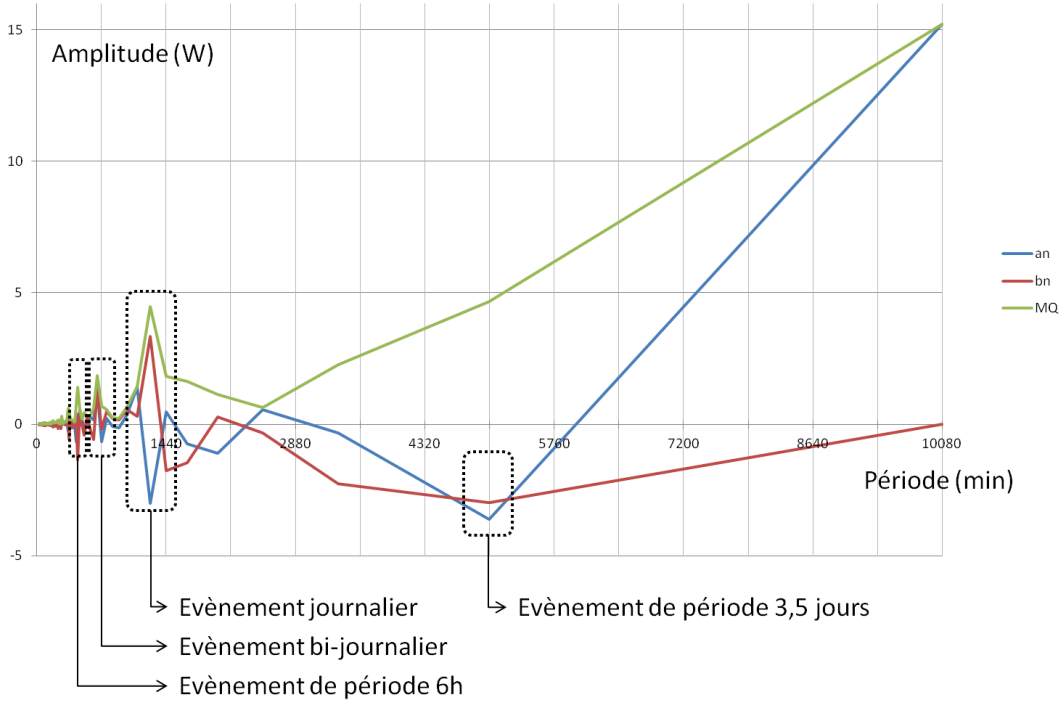


FIGURE 6.11 – Décomposition en séries de Fourier d'une courbe de charges, affichage des coefficients en fonction des périodes associées (An, Bn et la moyenne quadratique de An et Bn) (données SAVE)

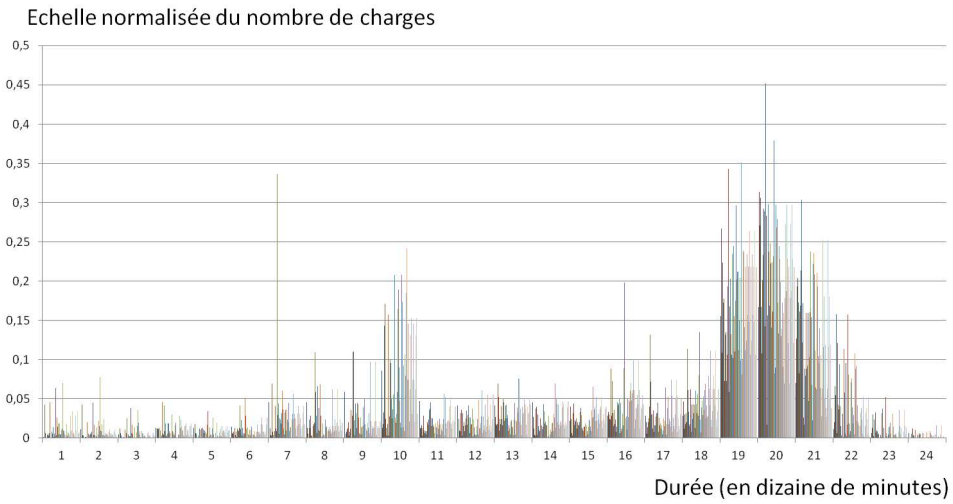


FIGURE 6.12 – Histogramme durée/fréquence, chaque couleur représente un utilisateur(données Kleber)

**4.2.1.5 Synthèse** Ces indicateurs permettent de réaliser un profil complet d'un utilisateur ou d'un véhicule, il en existe d'autres qui s'attachent, par exemple, au caractère géographique des charges. Il est utile de pouvoir caractériser un utilisateur et d'interpréter les indicateurs, toutefois pour grouper les utilisateurs il n'est pas possible d'utiliser tous les indicateurs à disposition :

- Les indicateurs globaux ne permettent pas de qualifier un utilisateur on ne peut donc pas les retenir pour les discriminer.
- Les courbes de charges journalières permettent d'agréger la durée des charges et le moment où elles ont lieu, on peut également définir les courbes de charges sur différents types de journées (jours ouvrés, jours fériés, vacances, etc.).
- Les histogrammes permettent d'isoler une information contenue dans les courbes de charges, ils sont redondants si des courbes de charges sont utilisées pour trier les utilisateurs.
- Les séries de Fourier, comme les histogrammes, isolent une information contenue dans une courbe de charges, elles ne seront pas utilisées en complément des courbes de charges.

Pour rappel, l'objectif des travaux entrepris étant, entre autres, de limiter les pics de consommation, il faut exploiter les indicateurs offrant la possibilité de détecter ces éventuels pics. Par conséquent les indicateurs relatifs aux fréquences ou spécifiques aux durées des charges ne sont pas les plus adaptés. Afin de proposer une classification des utilisateurs ce sont les courbes de charges journalières, au pas 10 minutes, qui ont été retenues car elles représentent tous les aspects d'une charge : moment de la journée, énergie et durée. Ces trois informations permettent d'obtenir des profils utilisateurs pertinents pour l'étude des pics de consommation.

### 4.2.2 Groupes d'utilisateurs

**4.2.2.1 Définition des groupes** Comme indiqué précédemment il n'est pas envisageable de traiter les utilisateurs au cas par cas c'est pourquoi on cherche à grouper les utilisateurs en catégories les plus homogènes possibles. L'objectif est de pouvoir apporter des solutions aux problèmes de pics de consommation à travers la formation de groupe d'utilisateurs pour s'affranchir des difficultés liées au traitement individuel. Selon l'analyse des indicateurs disponibles effectuée dans le paragraphe précédent c'est la courbe de charges journalière qui a été choisie comme indicateur discriminant.

**4.2.2.2 Apprentissage automatique** Pour former des groupes et donc simplifier le problème nous avons eu recours aux techniques d'apprentissage automatique (*machine learning*). Nous nous sommes placés dans le cas d'apprentissage non-supervisé [26], c'est-à-dire que l'on dispose de vecteurs à classer (les courbes de charges) sans à priori, ni connaissance sur leur appartenance à une classe particulière, le but est justement d'obtenir des classes homogènes.

L'algorithme retenu est celui des k-moyens (*k-means*) recommandé dans [27] pour le traitement des courbes de charges. Il fonctionne selon le principe suivant :

1. Sélection aléatoire de k (le nombre de classes souhaité) éléments pour former les premières classes.
2. Puis à chaque itération de l'algorithme les éléments sont rattachés à la classe la plus proche, c'est la distance au «milieu» de la classe qui est utilisée.
3. Les «milieux» de chaque classe sont recalculés avec les nouveaux éléments puis l'opération *2* est réitérée jusqu'à convergence, c'est-à-dire quand il n'y a plus de changement d'une itération à une autre.

Afin de pouvoir répéter cette classification avec l'arrivée de nouvelles données les «milieux» des classes sont conservés pour servir à initialiser le lancement de la nouvelle classification.

**4.2.2.3 Groupes obtenus** Pour exploiter les algorithmes de classification il est nécessaire d'indiquer le nombre de classes à obtenir. Ce travail a été réalisé avec les experts du domaine en comparant l'homogénéité et la représentativité des groupes obtenus en fonction du nombre de groupes.

Voici quelques résultats sur les groupes obtenus suite à la classification automatique des courbes de charges normalisées des utilisateurs de $\mathcal{VE}$. Sur les figures 6.13 et 6.14 chaque courbe représente la courbe de charges moyenne des utilisateurs du groupe.

Les groupes ont pu être identifiés en faisant intervenir différents experts de la $ME$ et en comparant les utilisateurs des groupes avec les utilisateurs dont on connait la nature (comme les particuliers) ou la nature du $\mathcal{VE}$ (comme les $\mathcal{VE}$ de fonction).

La figure 6.13 présente les courbes de charges moyennes des particuliers (*mParticulier*) et des utilisateurs de $\mathcal{VE}$ de fonction (*mFonction*). Les particuliers sont caractérisés par leurs charges nocturnes, quant aux utilisateurs de $\mathcal{VE}$ de fonction ils cumulent les charges nocturnes, comme pour les particuliers, et des charges sur le lieu de travail, ces dernières charges sont particulièrement régulières.

La figure 6.14 a nécessité d'avantage de croisement avec les autres disciplines de l'équipe du projet $\mathcal{VE}$ car il a fallu faire appel aux données de consommation, aux experts en mobilité et en sociologie. Ce travail a permis de mettre en avant différentes catégories parmi les utilisateurs de $\mathcal{VE}$ de service, c'est-à-dire les $\mathcal{VE}$ mis à disposition de plusieurs utilisateurs dans le cadre d'un usage professionnel. Le résultat de ce travail a permis de définir ces classes comme celles des utilisateurs autorisés à effectuer des trajets domicile-travail avec le $\mathcal{VE}$ (courbe *mServiceTDT*, Trajet-Domicile-Travail), des utilisateurs utilisant le $\mathcal{VE}$ le jour (*mServiceSoir)* et les véhicules partagés par plusieurs utilisateurs (*mService)* dont la courbe montre une mise en charge à tout moment de la journée.

**4.2.2.4 Comparaison avec l'approche sociologique** Au cours des expérimentations des études sociologiques ont été menées en parallèle des études des données de consommation ( [**?**], [102]). Ces études ont été réalisées sur la base de questionnaires et d'interviews des dif-

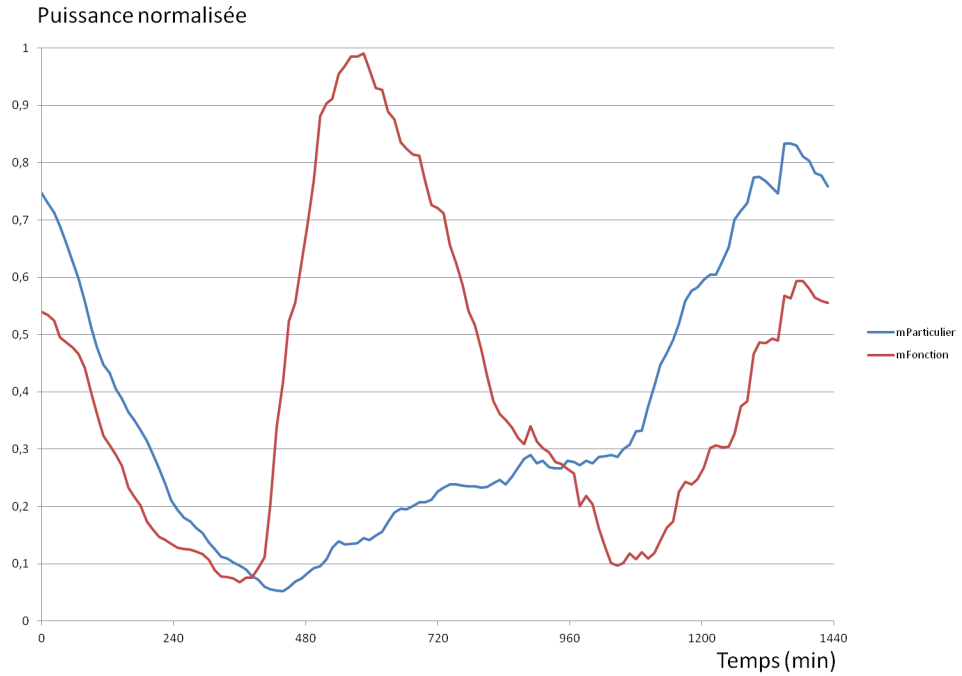


FIGURE 6.13 – Courbes de charges journalières normalisées moyennes des particuliers et des utilisateurs de véhicules de fonction (données SAVE)

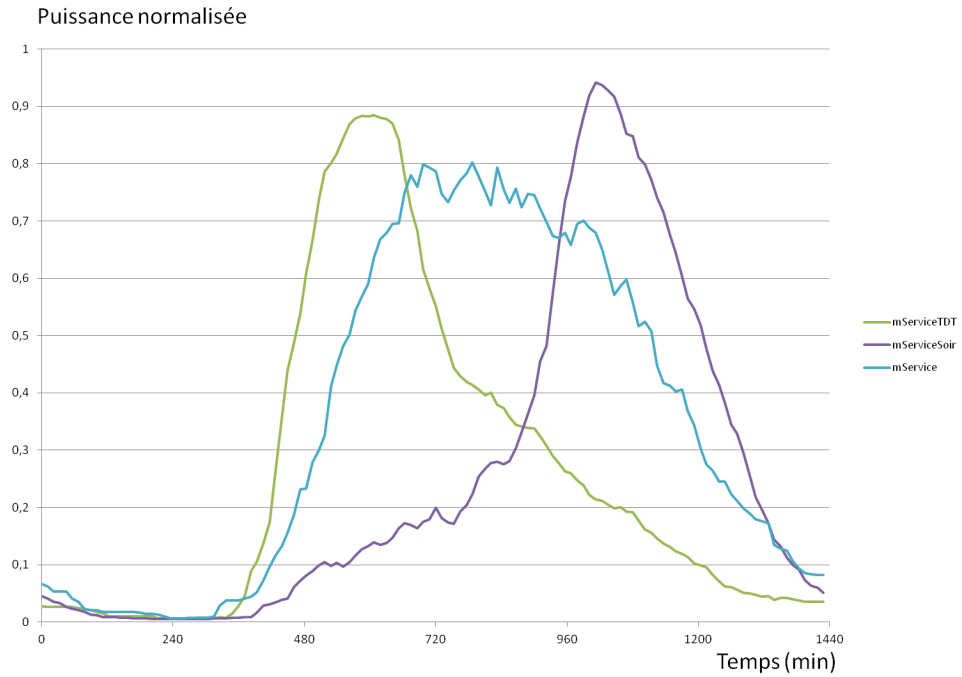


FIGURE 6.14 – Courbes de charges journalières normalisées moyennes des utilisateurs de véhicules de service (données SAVE)

férents participants, en combinant ces études sur la *ME* avec des études plus globales sur la mobilité des profils types de $\mathcal{VE}$ avaient été établis et par extension des profils d'utilisateurs. Pour chaque profil un type de consommation avait été établi sur la base d'éléments de mobilité connus pour donner la classification suivante :

- *Particulier* : individu possédant son propre $\mathcal{VE}$ pour réaliser des trajets personnels ainsi que ces trajets domicile-travail. Le profil attendu est caractérisé par une charge nocturne longue et des éventuelles charges plus courtes le jour.
- Les $\mathcal{VE}$ *de fonction* : comme pour le *particulier* on va retrouver des trajets domicile-travail et des trajets personnels à la différence que les charges vont principalement avoir lieu le jour et sur le lieu de travail.
- Les $\mathcal{VE}$ *de service* : ce type de véhicule va être utilisé le jour et mis en charge à deux moments privilégiés entre 12h et 14h et en fin d'après-midi. On distingue également les $\mathcal{VE}$ de service mono-utilisateur et multi-utilisateurs, le premier type est plus régulier que le second et dans le cas du multi-utilisateurs on observe de nombreuses charges courtes. Cette différence vient du manque d'information quant aux prochains utilisateurs, ce manque est lié à la nature de l'utilisation de ces $\mathcal{VE}$.

La confrontation de ces catégories avec les groupes obtenus a apporté des résultats intéressants. Les prévisions sur le comportement des particuliers se sont révélées tout à fait concordantes avec l'analyse des données, il en est allé de même pour le comportement des utilisateurs de $\mathcal{VE}$ de fonction. Tout comme la classification des groupes avait nécessité l'intervention des experts en sociologie sur la *ME*, la diversité au sein des utilisateurs de $\mathcal{VE}$ de services à pu être enrichie par l'analyse des données. Cela à conduit à différencier les utilisateurs de $\mathcal{VE}$ de services autorisés à réaliser des trajets domicile-travail des classes initialement définies.

Cette interaction entre des disciplines aussi diverses que la fouille de données, l'apprentissage automatique, la sociologie et la mobilité ont permis de comprendre les comportements et d'en définir les traitements (comme les classifications) de façon automatique grâce aux processus métiers.

### 4.2.3 Étude de la saisonnalité

**4.2.3.1 Hypothèse** La France fait face a des problèmes d'alimentation en électricité pendant des périodes particulièrement froides en hiver [23]. Ces problèmes peuvent entrainer des coupures de courant importantes dans certaines parties du territoire. Si à l'avenir la consommation des $\mathcal{VE}$ venait à s'ajouter à la consommation classique d'électricité ces problèmes se trouveraient aggravés. L'une des questions que se posent les décideurs est de savoir s'il faut mettre en place une solution spécifique à ces périodes pour les $\mathcal{VE}$. Si les comportements des utilisateurs de $\mathcal{VE}$ montrent des variations saisonnières, il faut alors mettre en place des solutions spécifiques en fonction des saisons. En revanche, si ce n'est pas le cas alors la gestion de ces périodes problématiques ne nécessite pas de solution particulière mais alors il faut une solution de gestion

23. http://www.cre.fr/reseaux/reseaux-publics-d-electricite/qualite-de-l-electricite#section8

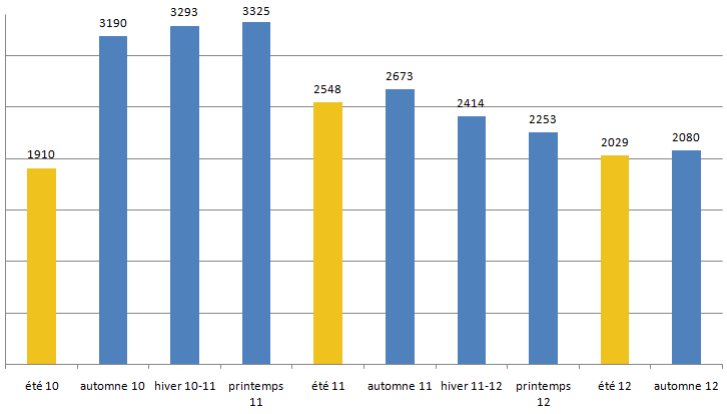


FIGURE 6.15 – Nombre de charges par saison, en jaune les étés pour aider la lecture (données Kleber)

générique.

L'étude de la saisonnalité cherche à tester l'hypothèse que les comportements sont saisonniers, cette hypothèse est soutenue par le constat que l'hiver les $\mathcal{VE}$ utilisent d'avantage de système auxiliaires (phares et conditionnement de l'habitacle). Il ne s'agit pas d'une analyse amenée à se répéter aussi fréquemment que celles présentées plus tôt dans ce chapitre toutefois la plate-forme est à même d'y apporter une réponse intéressante.

**4.2.3.2 Étude** Cette étude débute par une approche globale, il s'agit de calculer des indicateurs simples sur les charges. Comme on souhaite mesurer l'activité la première opération consiste à compter le nombre de charges réalisées durant chaque saison. Les données nécessaires sont directement disponibles, en l'occurrence il s'agit de compter des éléments de la table des faits. Nous avons enrichi cet indicateur en calculant la durée moyenne des charges comptées, les figures 6.15 et 6.16 illustrent les résultats obtenus sur l'étude des $\mathcal{VHR}$ de l'expérimentation Kleber. L'expérimentation a débuté à l'été 2010 d'où une faible activité à durant cette période. Dans la mesure où cette expérimentation s'est faite avec des moyens constants (nombre de $\mathcal{VHR}$ et nombre de bornes), on peut observer une baisse progressive de l'activité au fil des saisons. On n'observe pas de changements saisonnier dans le nombre de charges réalisées. On pourrait supposer que la durée de charge puisse varier avec les saisons reflétant ainsi le besoin et la consommation d'un supplément d'énergie. Cette hypothèse n'est pas vérifiée par les données disponibles (expérimentations Kleber et SAVE).

Pour caractériser l'activité par saison sans dépendre du nombre de charges réalisées, nous avons ré-exploité une connaissance déjà présente dans la plate-forme : les profils utilisateurs définis plus tôt.

D'abord, nous avons caractérisé les comportements saisonniers des utilisateurs par leurs courbes de charges journalières, chaque courbe de charges agrège les charges d'une saison.

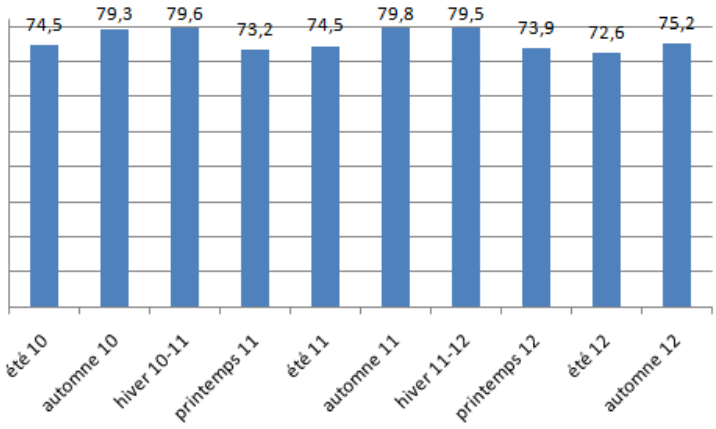


FIGURE 6.16 – Durée moyenne des charges en minutes par saison (données Kleber)

On stocke les profils obtenus pour chaque utilisateur et pour chaque saison, puis on compare systématiquement ces profils. Cette première analyse n'a pas montré de changement particulier d'une saison à une autre, ni de similitudes importantes entre les profils d'une même saison sur plusieurs années.

L'étude des utilisateurs au cas par cas ne permet pas de faire émerger de corrélations ou d'absence de corrélations en terme de comportements saisonniers. Pour mettre à jour des différences nous avons essayé d'agréger les profils d'une saison de façon à ajouter d'éventuels changements trop fins pour être observés au niveau des utilisateurs.

Dans la figure 6.17 chaque case indique le niveau de corrélation linéaire entre le profil d'une saison indiquée sur la ligne et celui d'une saison indiquée sur la colonne. Cet indice varie entre 0 et 1, plus il est proche de 1 et plus les éléments comparés sont similaires. On remarque que toutes les saisons présentent une forte corrélation avec n'importe quelle autre saison. Pour mieux visualiser les saisons les plus similaires nous les avons représentés par un dendrogramme [19] (voir figure 6.18) : plus le segment horizontal permettant de relier deux nœuds est bas et plus ces nœuds sont similaires.

On ne remarque aucun schéma significatif (les étés ne sont pas ensemble, les hivers et les étés n'ont pas tous le même degré de similitude ou de différence). Il n'y a donc pas de variation significative dans les profils d'une saison à une autre.

Le détail de cette étude a été diffusé comme document interne [114]. Compte tenu de ces études sur les données disponibles il n'est pas possible de conclure à une quelconque saisonnalité des comportements des usagers.

## 4.3 Intérêt de la plate-forme pour ces types d'analyses

Les connaissances, *i.e.* les analyses réalisées sur les données du domaine, sont générées automatiquement et stockées par la plate-forme. Cette répétabilité est soutenue par la capacité

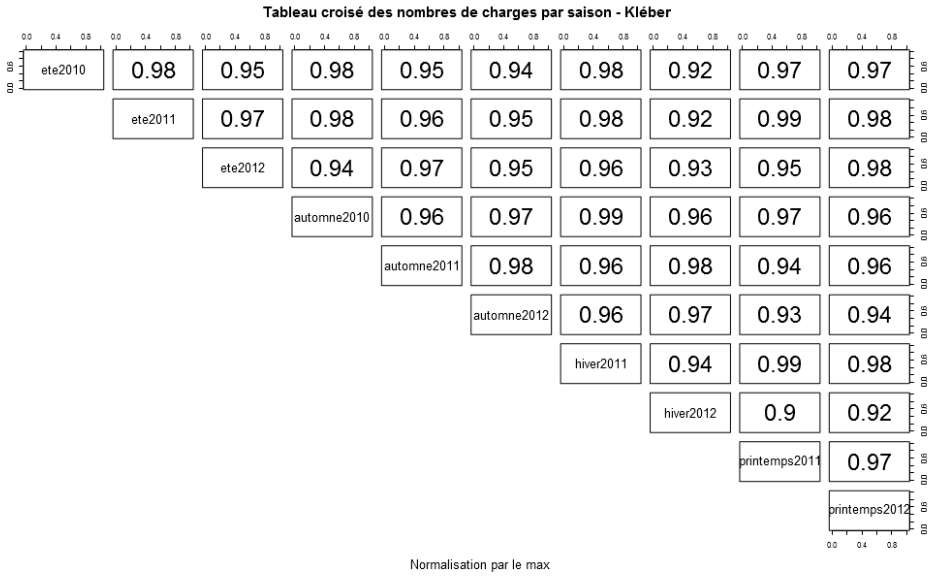


FIGURE 6.17 – Tableau des corrélations entre les courbes de charges journalières des saisons

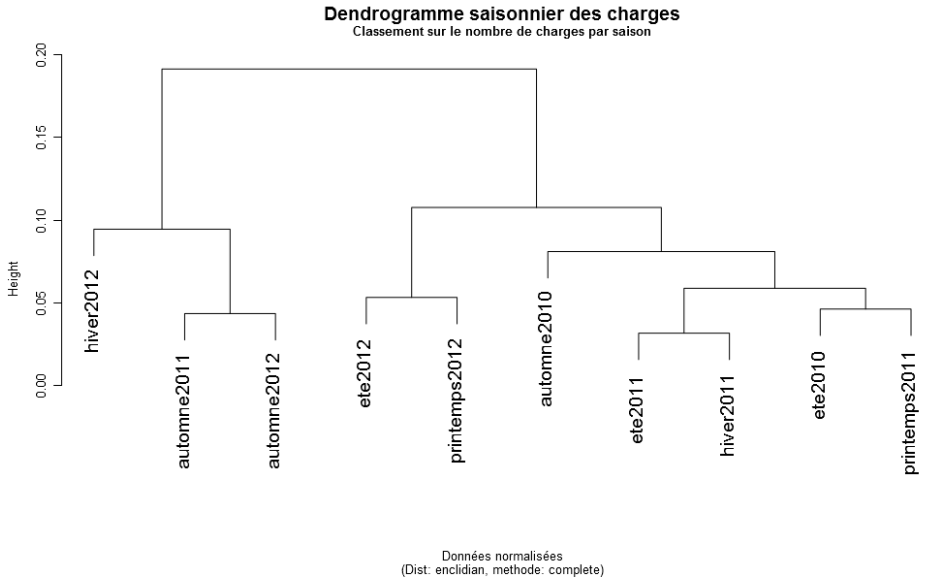


FIGURE 6.18 – Groupement des saisons selon leurs similitudes

à modifier aisément les processus métiers pour faciliter et rendre systématique le recours aux processus métiers. Une fois formalisée la connaissance devient facilement accessible et permet aux experts de se focaliser sur le traitement de la connaissance plutôt que sur sa génération.

La section suivante présente l'extension de ces analyses à l'étude des modèles d'affaires en se basant sur les concepts de la théorie des jeux.

# 5 Modèle d'affaires et théorie des jeux

## 5.1 Contexte et approche sans modèle

Jusqu'à présent les modèles de prévisions exploités par EDF sont généralement fournis par des données statiques d'infrastructures. En effet, EDF est capable de collecter les données de consommation d'un foyer ou d'un site industriel. Il s'avère que ces données forment des séries chronologiques quasi continues : les données sont générées 24h/24 et 7j/7 toutes les 10 minutes. Ces données de consommation sont relativement prévisibles en fonction de divers paramètres connus (saison, nombre d'occupants d'un logement, systèmes alimentés dans une usine, etc.).

Les données des charges issues des $\mathcal{VE}$ ne rentrent pas dans ce cadre car les consommations sont sporadiques dans le temps et dans l'espace. De ce fait la consommation des $\mathcal{VE}$ est moins généralisable que celle d'un bâtiment. Elle sera d'avantage liée à l'utilisateur ainsi qu'à des paramètres difficilement prévisibles tels que : les précipitations à un endroit donné, les embouteillages, les trajets imprévus, etc. Il ne paraît pas réalisable de modéliser finement le comportement de chaque utilisateur de $\mathcal{VE}$.

Ce constat nous a conduit à effectuer des analyses statistiques permettant de regrouper les utilisateurs dans des groupes les plus homogènes possibles. Fort de la connaissance de ces groupes et conscient que la modélisation de chaque utilisateur n'est pas envisageable nous proposons une **approche sans modèle**.

L'approche sans modèle signifie que les comportements, en l'occurrence ceux des utilisateurs de $\mathcal{VE}$, ne vont pas faire l'objet d'une modélisation. En revanche cette approche s'appuie sur l'historique complet des comportements. Ainsi pour anticiper les comportements cette approche va réaliser des traitements statistiques sur l'historique pour les prévoir.

Le concept de cette approche mise en œuvre est d'encourager les comportements les plus intéressants pour EDF grâce à des tarifs préférentiels pour fonctionner selon un modèle gagnant-gagnant pour l'utilisateur comme pour EDF :

- Pour l'utilisateur qui va évaluer l'économie réalisée par rapport à l'effort à fournir, par exemple le décalage de la charge d'une ou deux heures dans la soirée.
- Pour EDF qui va pouvoir consacrer son électricité à d'autres usages, l'industrie par exemple, et ainsi s'économiser l'achat d'électricité à un autre pays ou le démarrage d'une unité de production.

Il s'avère que cette façon d'aborder le sujet est caractéristique de la théorie des jeux. Aussi, nous nous sommes placés dans son cadre théorique pour formaliser notre méthode.

## 5.2 Intérêts et limites

L'intérêt réside dans le formalisme proposé par la théorie des jeux, celui-ci permet d'expliciter précisément les contraintes et les objectifs de l'approche sans modèle. La notion de formalisme est importante, car la théorie des jeux n'est pas un système de résolution de jeux à implémenter. La théorie des jeux offre des classifications des jeux et des approches pour résoudre certains types de jeux ou détecter des situations d'équilibre.

Cependant pour bien définir les joueurs il faut disposer d'un ensemble de données conséquent pour formaliser tous les types de comportements.

A partir de ce constat voici le formalisme retenu pour le cas d'EDF.

## 5.3 Cas d'EDF

Voici comment l'interaction entre EDF et ses clients utilisant des $\mathcal{VE}$ peut être formalisée.

Nous sommes dans le cas où les joueurs ne sont pas du même type. D'un côté il y a EDF et de l'autre des clients, chacun va avoir ses objectifs et sa façon de les évaluer, ses stratégies et ses réactions vis-à-vis des autres joueurs.

### 5.3.1 Joueur EDF

L'entité EDF est l'un des joueurs, le groupe EDF est le fournisseur d'électricité dans le *jeu*, la fourniture d'électricité est à la fois une obligation contractuelle source de revenus et un élément à réguler. Les perturbations de la fourniture d'électricité sont aujourd'hui liées à des demandes importantes des clients, elles arrivent typiquement en hiver en fin de journée lorsqu'il y a une demande forte en électricité.

Cette consommation permet de réaliser un gain financier direct (achat d'électricité). Mais si la consommation dépasse les capacités de production en activité alors EDF est contraint de démarrer des unités de production secondaires. Ces unités sont des centrales qui ne fonctionnent qu'en cas de dérégulation, c'est-à-dire quelques jours par an. Seulement elles demandent un entretien permanent, leur rapport énergie produite sur le coût d'entretien en fait des unités très coûteuses. Le coût de mise en marche de ces unités de production n'est pas équilibré la vente de l'électricité produite.

Il y a donc un intérêt très fort pour EDF à réguler la demande en énergie et éviter les perturbations, ceci afin de diminuer son parc d'unités secondaires coûteuses. Cette réflexion nous amène aux objectifs d'EDF concernant les $\mathcal{VE}$. Les études présentées dans les sections pré-

cédentes ont permis de former des groupes d'utilisateurs dont le comportement est homogène. Certains de ces groupes sont susceptibles de contribuer aux perturbations.

Ainsi l'objectif d'EDF sera d'encourager les utilisateurs de ces groupes à modifier leurs habitudes sur des périodes à risques. Le gain sera calculé en prenant en compte la baisse du prix de la fourniture en électricité des offres incitatives et les coûts évités grâce à ces offres. Les stratégies d'EDF vont donc s'orienter vers trouver les meilleures pratiques de prix dans une première approche.

### 5.3.2 Utilisateurs de $\mathcal{VE}$

Les utilisateurs de $\mathcal{VE}$ forment un type particulier de joueurs. En effet ils ne vont pas jouer les uns avec les autres mais uniquement avec EDF. Les études statistiques montrent qu'il est possible de créer des groupes de joueurs dont les comportements sont homogènes. Ce sont donc ces groupes qui seront les joueurs de notre étude.

Les intérêts des types d'utilisateurs sont très différents. Par exemple des particuliers seront attachés à la liberté de recharger chez eux à n'importe quel moment, un gestionnaire de flotte d'entreprise sera soucieux de disposer d'une partie de sa flotte chargée à n'importe quel moment à partir de 8h du matin, etc. L'interaction avec EDF est principalement financière, dans une moindre mesure certains utilisateurs attacheront une importance particulière à recharger lorsque la production est la plus écologique possible mais dans tous les cas l'intérêt sera la résultante de plusieurs facteurs, comme le montre *M. Pierre* [101] et que l'on retrouve dans l'étude sur la mobilité de *S. Le-Féon* [75].

### 5.3.3 Cadre retenu

Dans notre situation les joueurs sont les suivants :
- EDF dont l'objectif est d'éviter les pics de consommation d'électricité.
- Les groupes d'utilisateurs qui souhaitent pouvoir recharger selon leurs besoins de mobilité et probablement au meilleur prix.

La façon de procéder à ce «jeu» est la suivante :

1. EDF anticipe un pic de consommation.
2. EDF propose des offres incitatives aux groupes qui pourraient contribuer à ce pic afin de décaler leurs charges.
3. Les groupes concernés réagissent en fonction de leurs besoins.
4. EDF met à jour ses connaissances relatives aux réactions des groupes dans un entrepôt de données afin d'améliorer ses prochaines offres.

Il s'agit là d'un jeu séquentiel : les joueurs jouent chacun leur tour, et à information complète : les joueurs connaissent les possibilités d'actions, les gains et les motivations des autres joueurs.

## 5.4 Mise en œuvre et résultats

Les données disponibles sont relatives aux expérimentations présentées plus tôt, ces expérimentations ne suivaient pas de protocoles expérimentaux précis. De fait il n'y avait pas de contraintes exercées sur les comportements. Les données relevées sont donc uniquement représentatives du comportement naturel des utilisateurs, c'est-à-dire de leur façon d'utiliser leur $\mathcal{VE}$ indépendamment du coût de sa charge. Or le coût de la charge représente le principal levier d'EDF vis-à-vis des utilisateurs des $\mathcal{VE}$. Dans ce contexte il n'est pas possible d'estimer les gains des uns et des autres.

Toutefois la démarche a été menée à son terme pour plusieurs raisons :
- D'abord pour montrer la faisabilité d'une telle approche grâce à notre solution.
- Puis de montrer comment mobiliser la solution proposée dans cette thèse sur un cas opérationnel aussi pointu que celui de la tarification dynamique des charges des $\mathcal{VE}$ avec des contraintes de distribution de l'électricité.
- Et enfin de simuler cette approche en posant des hypothèses sur les comportements et ainsi d'en démontrer son intérêt.

**Démarche réalisée.** Pour montrer la faisabilité d'une telle approche des éléments manquants ont dû être simulés. En l'occurrence un niveau de satisfaction a été créé aléatoirement pour chaque utilisateur. L'objectif de l'utilisateur est d'atteindre ce niveau en acceptant les offres proposées par EDF et à les refuser si son niveau de satisfaction est plus haut que celui visé, ceci est illustré par un arbre de choix dans la figure 6.19.

Cette modélisation est grossière mais elle pourra être affinée par de nouvelles données. Par exemple si le prix et la production de $CO_2$ s'avèrent être les facteurs les plus importants pour les entreprises alors il faut définir la satisfaction comme une moyenne pondérée de ces critères pour les offres d'EDF. Le niveau cible de chaque utilisateur est à déterminer en fonction de ses acceptations et de ses refus.

Cette méthode a été implémentée sur la plate-forme, voici quelques figures illustrant le résultat de cette démarche sans modèle. La figure 6.20 montre la somme des courbes de charges journalières normalisées des groupes d'utilisateurs, chaque couleur représente l'énergie consommée par un groupe. Nous avons appliqué notre démarche sur les particuliers entre 21:30 et 23:30. Ce créneau a été choisi arbitrairement et nous avons choisi de cibler les particuliers uniquement. La figure 6.21 montre le résultat obtenu en terme de baisse de puissance électrique appelée sur le créneau indiqué. On observe une baisse de la consommation du groupe concerné sans toutefois l'annuler, en revanche la façon dont on simule l'offre montre un «*effet rebond*» après le créneau. Une fois sortie du créneau les charges ont été reprises par les utilisateurs pour conserver leur besoin de mobilité après la nuit.

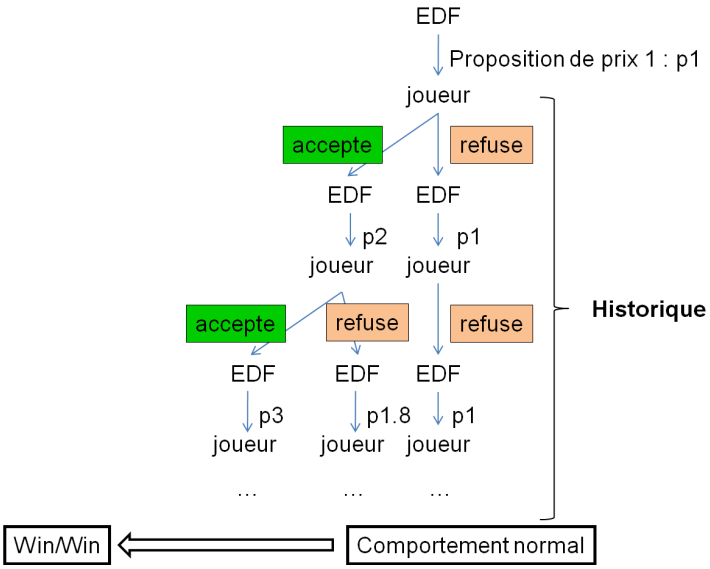


FIGURE 6.19 – Arbre des gains suivant les choix de l'utilisateur (l'offre pX est plus intéressante que l'offre pY pour X>Y)

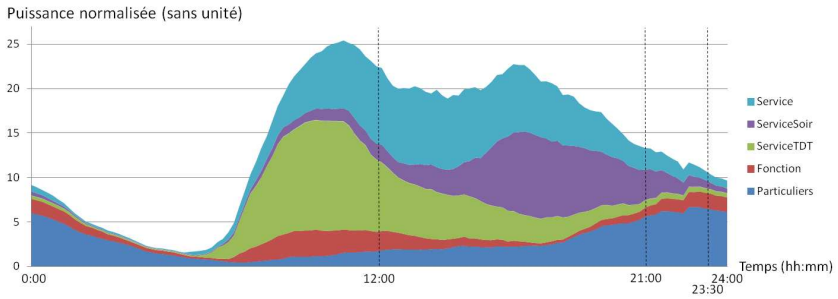


FIGURE 6.20 – Somme des courbes de charges journalières normalisées des groupes d'utilisateurs, les aires correspondent aux énergies consommées par chaque groupe (simulation basée sur les données SAVE)

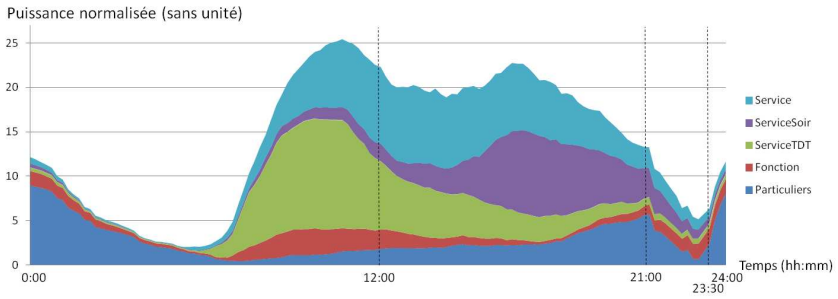


FIGURE 6.21 – Somme des courbes de charges journalières normalisées des groupes d'utilisateurs avec la mise en place de l'approche sans modèle sur les particuliers dans le créneau 21:30 - 23:30 (simulation basée sur les données SAVE)

## 5.5 Discussion

L'approche sans modèle est à double tranchant, d'un côté l'absence de modèle permet la mise en place immédiate de la solution, et des gains sont réalisés. D'un autre côté les gains ne sont pas prévisibles, toutefois cette méthode peut trouver sa place dans deux situations :

- Dans le cas d'un régime transitoire entre une solution empirique et une modélisation poussée des comportements afin de faire au mieux.
- Ou dans une phase de collecte de données afin de recueillir des données comportementales diversifiées à partir de contraintes réelles.

On remarque toutefois que l'approche sans modèle permet d'améliorer le fonctionnement «naturel» de la $ME$. L'approche sans modèle est aussi ajustable à de grands nombres de $\mathcal{VE}$, elle est donc susceptible d'être exploitée à l'avenir sur des parcs de $\mathcal{VE}$ conséquents.

# 6 Conclusion

L'exploitation de l'ingénierie de données pour la $ME$ est une évolution naturelle pour une activité qui a toujours été à la pointe de la technologie.

La solution que nous avons proposée a fait ses preuves au sein du projet «Véhicules Électriques». D'abord pour collecter les données puis pour les analyser. Comme nous l'avons montré il est possible de réaliser un vaste éventail d'analyses.

Ces analyses couvrent la gestion courante des éléments du domaine, comme la surveillance des infrastructures et la détection des pannes. Mais aussi la génération d'informations plus complexes sur les utilisateurs et leurs habitudes.

Enfin la flexibilité de notre solution permet d'amener l'étude d'un domaine technique plus loin en permettant à la plate-forme d'interagir dynamiquement avec le domaine. Nous avons mis en œuvre cette interaction sur la base d'un modèle d'affaire dynamique formalisé par la théorie des jeux.

A travers ces travaux nous avons pu démontrer que la solution proposée répond aux problématiques soulevées lors de l'initiation de ces travaux.

Chapitre 7

# Conclusions et perspectives

## Sommaire

# 1 Synthèse de la démarche

Cette synthèse de la démarche entreprise rappelle tout d'abord les objectifs assignés aux travaux, puis présente la solution proposée, et souligne enfin l'aspect générique de la solution.

## 1.1 Rappel des objectifs industriels

EDF disposait de nombreuses sources de données relatives au domaine de la mobilité électrique ($ME$), ces sources traitaient principalement des charges des utilisateurs de véhicules électriques ($\mathcal{VE}$). Cependant ces sources présentaient, et présentent toujours, une forte hétérogénéité. Cette hétérogénéité se retrouve dans les formats, les modèles de données, les supports des données, la sémantique, etc. Ainsi, avant toute chose le premier objectif était d'apporter une solution pour uniformiser ces données afin d'en faciliter utilisation. Pour que cette solution ne reste pas un correctif local mais puisse amener à d'avantage d'uniformité, un objectif secondaire a été défini : il fallait que la solution proposée amène des éléments d'uniformisation à partager avec les différents partenaires impliqués dans la $ME$. Cet objectif est capital pour un meilleur fonctionnement de la $ME$ sur le long terme.

En parallèle des sources de données sur le domaine, EDF disposait d'experts sur différentes parties du domaine, ces derniers représentaient une mine de processus, ou connaissances, métiers. Pour exploiter ces processus avec les données, des solutions ont été développées par EDF. Malgré les difficultés rencontrées pour exploiter ces processus lors de la manipulation des premières données (*i.e.* : adaptation à des nouveaux jeux de données, répétabilité, etc.) des solutions ont été développées pour les analyser. Ces solutions souffraient des problèmes d'hétérogénéité mais elles présentaient également des limites en termes d'analyse, de volume de données, de partage et de ré-utilisation. Nous reviendrons sur les solutions mises en place dans la suite de ce chapitre pour les comparer à la solution proposée. Cet état de fait a amené le second objectif de la thèse : la solution proposée devait augmenter les capacités d'analyse déjà existantes, c'est-à-dire permettre des rapports plus complets, actualisés plus souvent, capables de trier les données urgentes (signaler les pannes), supporter des traitements statistiques poussés, etc.

Enfin l'objectif *commercial* de ces travaux relève de l'exploitation des connaissances sur les utilisateurs pour améliorer la fourniture d'électricité. C'est-à-dire minimiser la contribution de la $ME$ aux pics de consommation et offrir une capacité à moduler la consommation des utilisateurs. La théorie des jeux a été testée pour réaliser ces objectifs, et la solution proposée aux objectifs précités devait permettre l'intégration d'un tel système de pilotage.

Souvent ce genre de demande est directement sous-traitée par EDF, en effet de nombreuses sociétés proposent des solutions d'entreposages, de maintenance de serveurs, etc. Or EDF souhaite maîtriser la solution en interne pour s'affranchir de toutes dépendances, monter en compétences et disposer d'une solution sur mesure rapidement modifiable par les différents experts. Cette volonté a également apporté des nouvelles contraintes, la solution proposée ne s'adresse

en effet pas à des experts en informatique mais à des experts de multiples domaines qu'il n'est pas envisageable de former à toutes les techniques d'entreposage des données habituellement requises.

## 1.2 Solution proposée

Notre solution se situe à l'intersection de plusieurs disciplines, à commencer par l'intégration de données, l'entreposage, les ontologies, les processus et la théorie des jeux. Très peu de travaux couvrent l'intégralité du cycle de vie d'une entrepôt de données ($ED$) en revanche les différentes communautés ont apporté une multitude de solutions à certaines étapes du cycle de vie pour : l'intégration, le stockage, l'optimisation, etc. Or si l'on synthétise les objectifs d'EDF on retrouve un besoin pour un cycle de vie complet. Nos travaux tentent de répondre à ce besoin spécifique.

D'abord nous nous sommes attachés à la création d'une ontologie dans un contexte aussi contraint que celui de la recherche et du développement en entreprise sur des sujets innovants, évoluant rapidement et où les solutions doivent être opérationnelles immédiatement. La méthode que nous avons proposée a permis la mise en place d'une ontologie sur le domaine de la $ME$, en vue d'exploiter les travaux relatifs au stockage de données à base ontologique.

A partir d'une telle ontologie nous avons détaillé la méthode de conception d'un entrepôt de données à base ontologique ($EDBO$), pour cela nous nous sommes basés sur les travaux liés à l'intégration des données, comme par exemple les algèbres pour les processus $ETL$. Nous avons proposé une solution à partir d'un $ETL$ sémantique capable de s'adapter à de nouvelles sources décrites par l'ontologie ainsi qu'à des changements dans l'ontologie et dans le modèle de données de l'$EDBO$.

A partir de l'$EDBO$ destiné aux données du domaine nous avons conçu un niveau plus abstrait avec les connaissances. Cette sur-couche à l'$EDBO$ permet de décrire la connaissance générée ou à générer de telle manière qu'elle constitue un mode d'emploi à la création et la manipulation des connaissances. Elle sert également de système d'information de la connaissance pour faciliter la capitalisation des connaissances et favoriser leur partage. Pour répondre aux derniers objectifs (gestion des processus et objectif commercial) les processus métiers ont été intégrés à cette couche. Ils rentrent dans la description des connaissances et permettent de générer la connaissance automatiquement en exploitant le formalisme proposé par la *Business Process Modelisation and Notation*.

Pour résumer, notre solution repose sur une ontologie de domaine, construite selon une nouvelle méthode, sur un entrepôt de données à base ontologique et une ontologie de connaissances accompagnée des processus métiers. Ainsi à partir de trois éléments déclaratifs, les ontologies et les processus métiers, notre solution couvre l'intégralité du cycle de vie des données, comme l'illustre la figure 7.1. Pour démontrer que tout le cycle est couvert nous avons implémenté une solution de pilotage de la consommation à partir de la théorie des jeux pour proposer un système

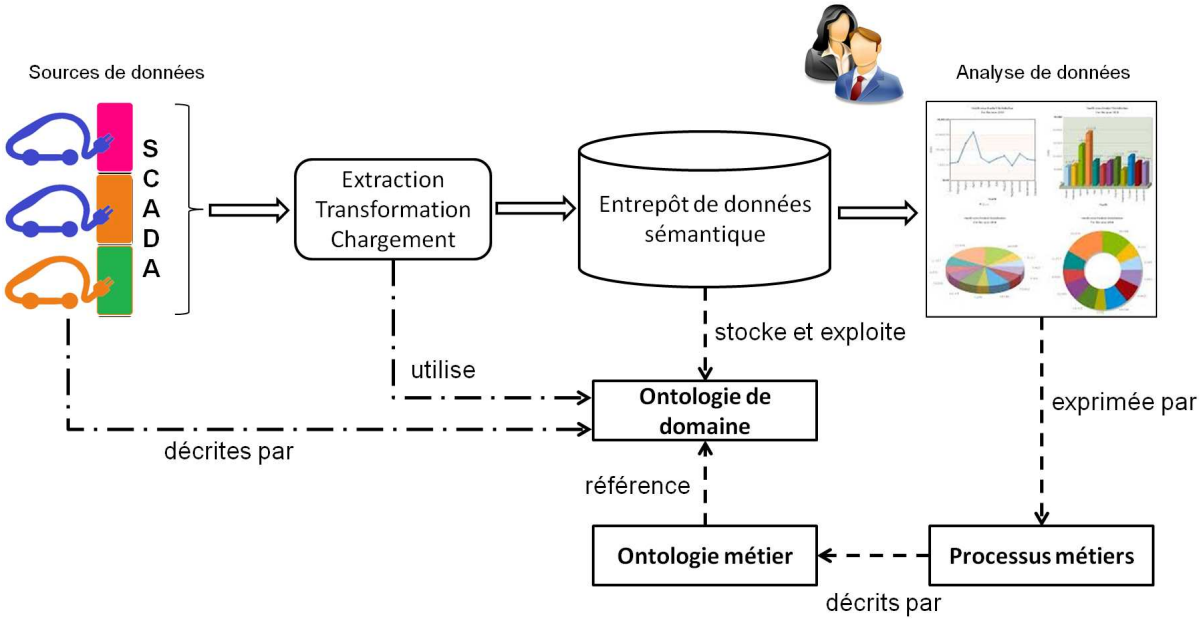


FIGURE 7.1 – Gestion complète du cycle de vie

autonome.

### 1.3 Méthode générique et globale

Cette solution a été développée pour les besoins d'EDF concernant la *ME*, cependant comme le montre le rappel des objectifs ces besoins sont courants dans l'industrie. Uniformiser et homogénéiser les données sont des tâches récurrentes dans les systèmes d'information et ce sont des sujets de recherche actuels. La question de l'intégration des données sans avoir à réaliser, ou à faire réaliser, des développements coûteux est un but poursuivi par les chercheurs comme par les industriels. Enfin le traitement et la maîtrise de la connaissance représentent une partie importante de l'avenir de l'économie.

Les méthodes créées, et testées, dans cette thèse sont applicables à d'autres domaines où ces besoins existent.

## 2 Synthèse des résultats

Après cette synthèse relative à la solution proposée, intéressons nous maintenant aux résultats obtenus, en commençant par une comparaison avec les méthodes précédemment employées.

| Méthode | Difficulté de mise en place | Temps de mise place | Volume de données | Flexibilité | Évolutivité | Gestion de la connaissance |
|---|---|---|---|---|---|---|
| Tableur | Facile | Rapide | limité | aucune | faible | aucune |
| *BDD* dédiées | Moyen | Moyen | important | limitée | moyenne | correcte |
| *BDD* unique | Moyen | Moyen | important | correcte | moyenne | correcte |
| Entrepôt | Difficile | Long | très important | correcte | bonne | correcte |
| *EDBO* | Moyen | Moyen | très important | bonne | bonne | bonne |

TABLE 7.1 – Tableau de comparaison des solutions

## 2.1 Comparaison avec les solutions précédentes

Le tableau 7.1 offre un aperçu des indicateurs utilisés pour évaluer les solutions de gestion des données (les *BDD* dédiées correspondent à la mise en place d'une *BDD* et d'un $ETL$ par source de données).

Les solutions à base de tableurs, dotées de fonctions avancées (matrices, macros, etc.), ont été les premières à être mises en place. Elles sont rapides à mettre en place, toutefois elles ne permettent pas de gérer des volumes de données importants. De plus elles sont dédiées à un jeu de données précis, ne sont pas flexibles (changement de format, colonnes supplémentaires, etc.) et sont faiblement évolutives. De surcroit, la gestion des connaissances n'est pas prise en compte et le partage est délicat (chacun possède sa version).

Les bases de données (*BDD*), plus complexes à mettre œuvre, permettent virtuellement de s'affranchir des problèmes de volumes. Elles sont davantage flexibles et évolutives, au prix du re-développement de l'$ETL$ et des outils d'analyse, et la gestion de la connaissance est possible. La mise en place d'une solution type *BDD* sur un serveur offre des possibilités de partage du contenu.

Un *ED* permet de gérer encore davantage de données mais sa mise en place fait appel à des techniques plus complexes. Cependant la possibilité d'ajouter des dimensions donne des possibilités d'évolutions plus importantes qu'une *BDD*.

La solution que nous proposons possède la même complexité de mise en œuvre que les solutions de type *BDD* mais pour des capacités bien plus grandes.

Les paragraphes suivants résument les principaux résultats obtenus.

## 2.2 Application des processus métiers

### 2.2.1 Usage courant

La plate-forme permet aujourd'hui d'intégrer rapidement toutes nouvelles sources de données grâce à la flexibilité de l'ontologie et ces données peuvent être immédiatement traitées.

La plate-forme développée offre en premier lieu un accès direct aux données, cela permet de superviser le domaine. Cette supervision permet la surveillance du matériel afin d'en détecter le mauvais fonctionnement ainsi que de rendre compte de l'activité de la *ME* pour alimenter, par exemple, les études sociologiques sur les déplacements et les habitudes des utilisateurs ou les études sur l'impact de la *ME* sur le réseau de distribution.

Au delà de la supervision, l'accès aux données du domaine donne la possibilité de réaliser des études *à usage unique* sur les données. Cela est rendu possible par l'utilisation directe des termes ontologiques, donc définis par les utilisateurs, dans les requêtes.

Sans la partie connaissance, l'usage courant est déjà possible, il sera amélioré et automatisé par la couche connaissance de la plate-forme.

### 2.2.2 Gestion de la connaissance

La partie connaissance de la plate-forme a permis d'amener l'usage courant à un autre niveau. Cette amélioration a été rendue possible par une description facilitée et intégrée à la plate-forme des connaissances et de leurs méthodes de génération (les processus métiers).

**2.2.2.1 Ontologie des connaissances** Cette ontologie prolonge celle du domaine, toutefois cette dernière n'est pas partagée en dehors d'EDF car elle rassemble et décrit l'expertise des experts de la *ME*. Construite de la même façon que l'ontologie de domaine elle dispose des mêmes qualités. Ainsi les concepts à traiter comme des tables de faits sont déployés comme les faits du domaine par la méthode décrite dans le chapitre 4. Les concepts de l'ontologie des connaissances sont reliés à ceux de l'ontologie de domaine pour permettre de parcourir les ontologies et d'exploiter ces relations soit pour comprendre comment est calculée une connaissance, soit pour voir les connaissances issues d'un concept du domaine. Enfin les processus métiers représentent le bras armé de cette ontologie en permettant la génération automatique des instances des concepts de l'ontologie des connaissances, comme indiqué dans le paragraphe ci-dessous.

**2.2.2.2 Processus métiers** Comme mentionné à plusieurs reprises, la valeur de la solution est aussi estimée par rapport au temps des experts qui ne sera pas dépensé sur des tâches fastidieuses comme : nettoyer les données, les intégrer, refaire des analyses déjà mises au point par d'autres experts, etc. L'ontologie de domaine et l'*EDBO* ont permis de prendre en charge automatiquement le nettoyage et l'intégration des données. L'intégration des connaissances au sein de la même plate-forme que l'*EDBO* autorise une transmission et un partage de la connaissance accrus :

- La description de la connaissance sous forme d'ontologie constitue un mode d'emploi pour la générer.
- Cette description est reliée à l'ontologie de domaine afin de pouvoir, par exemple, parcourir les concepts du domaine et observer quelles sont les connaissances qui s'y rattachent,

et réciproquement.
- Les connaissances déjà générées sont disponibles au même titre que les données dans la plate-forme.

Ce qui nous amène au dernier point offrant des facilités aux experts : les processus métiers. Ils viennent compléter la description des connaissances et capitalisent le savoir des experts. Les processus métiers tels que nous les avons représentés respectent un formalise rigoureux qui permet leurs exécutions par la plate-forme. L'exécution des processus métiers génère des connaissances qui viennent peupler l'entrepôt des connaissances.

Ainsi la plate-forme avec ces trois éléments déclaratifs que sont les ontologies (de domaine et des connaissances) et processus métiers permet aux experts de travailler moins sur les tâches répétitives. A partir de là les experts peuvent se focaliser sur la manipulation de la connaissance pour créer de la valeur, comme : créer des services et des offres commerciales, mener des études sociologiques approfondies pour mieux comprendre la $ME$, etc.

La section ci-dessous propose de synthétiser quelques résultats obtenus à travers la plate-forme.

### 2.2.3 Analyses des comportements

Afin de démontrer la faisabilité et l'efficacité de notre solution plusieurs études ont été menées pour contribuer à la connaissance de la $ME$ par les équipes d'EDF. L'étude des données a été ensuite comparée aux études sociologiques et de mobilité, le croisement de plusieurs disciplines a permis d'apporter des éclairages nouveaux et d'enrichir l'expertise d'EDF.

Les données disponibles ont ainsi permis de valider des catégories d'utilisateurs et de mettre en évidence des subtilités à l'égard de certains groupes. Comme cela était attendu, les particuliers réalisent l'immense majorité de leurs charges à domicile. L'analyse des données a permis de quantifier ce comportement et les analyses sociologiques ont permis d'identifier certains motifs à ce comportement, notamment l'aspect pratique (prise disponible, rangement du câble, sécurité, etc.). Les utilisateurs de véhicules de service, initialement divisés en fonction du caractère mono-utilisateur ou multi-utilisateurs des véhicules, font apparaître un nouveau paramètre discriminant : la possibilité de réaliser des trajets domicile-travail. Ce résultat a son importance car le comportement change de façon importante en fonction de ce paramètre et les offres commerciales qui seront proposées en tiendront compte.

Le caractère saisonnier des comportements faisait partie des indicateurs attendus pour le dimensionnement des offres commerciales. Cette attente vient de l'augmentation et des pics de consommation observés l'hiver et la crainte que la $ME$ ne vienne aggraver ces effets. Toutefois les données disponibles ne montrent pas de changement dans les habitudes des utilisateurs au cours des différentes saisons. En revanche, les congés des utilisateurs ont un impact sur la consommation de la $ME$ : on observe ainsi des baisses de fréquence des charges au moment des vacances scolaires.

Ces exemples ont permis d'illustrer l'efficacité de la solution mise en place pour traiter des problèmes complexes. La section ci-dessous expose une possibilité d'extension de la solution à du pilotage du domaine.

### 2.2.4 Utilisation de la théorie des jeux pour la création d'un modèle d'affaire

La *ME* n'est pas encore finement encadrée par des offres tarifaires de part sa nature complexe (différenciation du véhicule, de l'utilisateur et de l'infrastructure de charge). L'approche par la théorie des jeux s'attache à proposer un système tarifaire destiné aux utilisateurs sans nécessiter une modélisation précise des utilisateurs, ce cas de figure correspondrait à de l'optimisation.

La première étape a consisté à récupérer les classes d'utilisateurs établies dans les études précédentes. Ce sont les classes qui seront considérées comme les joueurs. Ce choix est justifié par le degré de connaissance plus élevé des classes, donc de l'agrégation d'utilisateurs, que le degré de connaissance de chacun des utilisateurs. En simulant un critère, amené à être connu à l'avenir, nous avons pu simuler l'interaction automatique de la plate-forme avec les groupes d'utilisateurs.

Suivant le besoin d'EDF, relatif à la production et à la consommation d'électricité, la plate-forme possède plusieurs stratégies pour obtenir des résultats. Cette interaction est basée sur le degré de satisfaction des utilisateurs, son but est de faire coïncider la satisfaction des utilisateurs avec les intérêts d'EDF. La plate-forme pilote alors les offres et étudie ensuite les réaction des utilisateurs pour affiner sa connaissances des groupes et de leurs réactions. Afin de modifier suffisamment le comportement des utilisateurs, la plate-forme va tenter une stratégie qui répond à leurs besoins tout en minimisant la contre-partie qu?EDF doit leur concéder Pour pérenniser ce système les offres émises visent à satisfaire les utilisateurs et à récompenser ceux qui répondent favorablement aux offres pour créer une situation gagnant-gagnant.

La théorie des jeux permet donc de réaliser une **approche sans modèle de comportements des utilisateurs en se basant sur du *feedback*, c'est-à-dire l'analyse des réactions pour apprendre**. Notre solution vise à fournir une démarche complète d'intégration des données et des processus au niveau sémantique, d'automatisation des processus, enfin la théorie des jeux nous permet d'achever ce cycle sur une action sur le domaine dans un but économique.

# 3 Conclusion et perspectives

## 3.1 Conclusion

### 3.1.1 Approche complète

La solution proposée à EDF a été bien accueillie, elle couvre le cycle de vie de la donnée depuis son intégration jusqu'à son traitement à des fins opérationnelles à partir d'éléments descriptifs (des ontologies et les processus métiers). Les besoins clairement exprimés ont permis d'utiliser les derniers travaux de la communauté de recherche et de proposer de nouvelles idées. Le résultat obtenu avec la plate-forme *OntoDB* est la preuve que les travaux académiques menés sont en phase avec les besoins des industriels.

### 3.1.2 Viabilité économique

Les résultats obtenus ont permis de démontrer l'efficacité de la solution et ils ont contribué à enrichir la connaissance d'EDF sur la $ME$. Les premiers éléments de la solution proposée, que sont la nouvelle méthode de création d'ontologie et la mise place (conception et déploiement) d'un entrepôt de données sémantique, bien que requérant la prise en main d'outils sémantiques, s'avèrent moins coûteux (en temps) que les solutions précédemment mises en place. Ce premier avantage est primordial vis-à-vis d'un industriel, il nous a ensuite permis d'apporter de nouvelles contributions sur la gestion des connaissances. La gestion des connaissances et la mise au point des processus métiers, au sein de la même plate-forme comme une sur-couche de la partie donnée, a abouti à une exploitation réussie des données du domaine pour l'usage courant (facturation, surveillance du réseau) et l'usage prospectif (comportements et habitudes des utilisateurs).

### 3.1.3 Support pour de nouvelles approches

La maîtrise du cycle de vie des données que nous avons proposée permet de mettre à la disposition des experts l'intégralité des données disponibles selon un langage qu'ils maîtrisent. Dès lors différents types d'approches peuvent être testés facilement. Comme par exemple la théorie des jeux qui représente une approche économique viable se basant sur l'apprentissage du domaine. Cette approche pour piloter le domaine n'est pas nécessairement meilleure par rapport à de l'optimisation (possible lorsque l'on connait bien le domaine), mais elle constitue une amélioration par rapport à une absence de pilotage.

### 3.1.4 Méthode générique

Nous disposons à présent d'une méthode globale et générique validée par un cas d'étude répondant à des problématiques classiques permettant d'étudier un domaine. Nous avons proposé une démarche générique à base ontologique qui pourra être instanciée à d'autres cas d'étude. Ce résultat est encourageant quant à l'intérêt des industriels pour les différentes communautés de recherche dont nous avons étudié et exploité les travaux.

Cette méthode peut également servir de base modifiable pour tester des nouveaux éléments (comme le calcul de nouveaux indicateurs sur le domaine) et supporter des applications critiques, comme des politiques tarifaires.

## 3.2 Perspectives

### 3.2.1 Développements

Les perspectives à court terme relèvent de l'amélioration de la plate-forme. En premier lieu il reste des concepts de la notation *BPMN* à intégrer à la plate-forme de façon à étoffer les processus métiers et permettre l'exécution de processus plus complexes. Le deuxième élément à développer à court terme est une interface graphique pour faciliter l'accès à la plate-forme au plus grand nombre (modifier les ontologies et les processus métiers) et de former les experts aux techniques de création et de manipulation d'ontologies.

Parmi les développements on peut également indiquer le travail de diffusion de l'ontologie de domaine qui doit être poursuivi pour que des services inter-partenaires puissent être étendus plus rapidement, par exemple : des possibilités de parking (supermarchés, cinémas, etc) et de charges, des stations de recharge qui diffusent leurs disponibilités aux utilisateurs à proximité, etc.

### 3.2.2 Travaux de recherche

Les contributions présentées dans cette thèse ont été entreprises avec la volonté de fournir une solution transversale pour couvrir le cycle de vie du traitement de la donnée avec des entrepôts de données.

L'utilisation qui est faite dans notre solution des ontologies n'exploite pas tout leur potentiel et plus précisément la capacité à réaliser des inférences. La plate-forme et les utilisateurs bénéficieraient énormément d'une capacité à relier automatiquement des données : faire correspondre des badges utilisateurs à des véhicules pour permettre une meilleur analyse des comportements, boucher des «trous» dans les données provenant de certaines sources, etc.

De plus, nous effectuons des vérifications manuelles (comme vérifier la cohérence de l'ontologie) qui pourraient peut-être être automatisées par des méthodes formelles.

### 3.2.3 Et au delà ?

A partir des problématiques identifiées par EDF pour ses propres besoins nous avons bâti une solution générique qui répond en soi aux problématiques efficacement. Mais cette solution ouvre également la voie à des travaux plus larges, à ce titre la plate-forme correspond à un squelette capable d'accueillir des modules comme nous l'avons fait avec la gestion des connaissances. Il devient dès lors possible d'intégrer des travaux existants à cette plate-forme, comme ceux cités dans l'état de l'art sur l'utilisation des ontologies pour la définition des besoins ou pour optimiser le traitement des données. Et surtout, elle peut servir de base pour tester des travaux sur les ontologies, les entrepôts, les processus ou des concepts économiques et sociologiques, et à présenter l'intérêt de ces travaux à des industriels comme EDF.

# Bibliographie


[1] SWAD-Europe Deliverable 10.2: Mapping Semantic Web Data with RDBMSes.

[2] ESPRIT, the European Strategic Programme for Research and development in Information Technology. Technical report, 1990.

[3] Les Français prêts pour la voiture électrique ? *Energie Plus*, 533, October 2014.

[4] D.J. Abadi, A. Marcus, S. Madden, and K. Hollenbach. Sw-store: a vertically partitioned dbms for semantic web data management. *VLDB Journal*, 18(2):385–406, 2009.

[5] Christopher Adamson. *Mastering data warehouse aggregates: solutions for star schema performance*. Wiley Pub, Indianapolis, IN, 2006.

[6] Sofia Alexaki, Vassilis Christophides, Gregory Karvounarakis, Dimitris Plexousakis, and Karsten Tolle. The ics-forth rdfsuite: Managing voluminous rdf description bases. In *Proceedings of the 2nd International Workshop on the Semantic Web (SemWeb 2001)*, 2001.

[7] Nathalie Aussenac-Gilles, Davide Buscaldi, Catherine Comparot, and Mouna Kamel. Enrichissement d'ontologies grâce à l'annotation sémantique de pages web. In *EGC*, pages 229–234, 2013.

[8] Chuck Ballard, Dirk Herreman, Don Schau, Rhonda Bell, Eunsaeng Kim, and Ann Valencic. *Data modeling techniques for data warehousing*. IBM, 1998.

[9] Len Bass, Paul Clements, and Rick Kazman. *Software architecture in practice*. Addison-Wesley, Reading, Mass., 1998.

[10] C Bauzer-Medeiros, O. Carles, F Devuyst, B Hugueney, M Joliveau, G. Jomier, M. Manouvrier, Y. Naïja, G. Scemama, and L Steffan. Vers un entrepôt de données pour le trafic routier. February 2006.

[11] Sonia Bergamaschi, Francesco Guerra, Mirko Orsini, Claudio Sartori, and Maurizio Vincini. A semantic approach to etl technologies. *Data & Knowledge Engineering*, 70(8):717–731, 2011.

[12] Mike Bergman. A Brief Survey of Ontology Development Methodologies, August 2010.

[13] Nabila Berkani, Ladjel Bellatreche, and Selma Khouri. Towards a Conceptualization of ETL and Physical Storage of

Semantic Data Warehouses as a Service. 2013.

[14] Donald J. Berndt, Alan R. Hevner, and James Studnicki. The Catch data warehouse: support for community health care decision-making. *Decision Support Systems*, 35(3):367 – 384, 2003.

[15] BetterPlace. Better Place, 2013.

[16] Barry W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, 1988.

[17] Ilyes Boukhari. *Intégration et exploitation de besoins en entreprise étendue fondées sur la sémantique*. PhD thesis, ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechique-Poitiers, January 2014.

[18] Erol Bozsak, Marc Ehrig, Siegfried Handschuh, Andreas Hotho, Alexander Maedche, Boris Motik, Daniel Oberle, Christoph Schmitz, Steffen Staab, Ljiljana Stojanovic, Nenad Stojanovic, Rudi Studer, Gerd Stumme, York Sure, Julien Tane, Raphael Volz, and Valentin Zacharias. KAON - Towards a Large Scale Semantic Web. In *Proceedings of the 3rd International Conference on E-Commerce and Web Technologies (EC-WEB'02)*, pages 304–313, London, UK, 2002. Springer-Verlag.

[19] Leo Breiman. *Classification and regression trees*. Chapman & Hall, New York, N.Y., 1993.

[20] Dan Brickley and R. V. Guha. *RDF Vocabulary Description Language 1.0: RDF Schema*. World Wide Web Consortium, February 2004.

[21] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, ISWC '02, pages 54–68, London, UK, UK, 2002. Springer-Verlag.

[22] Robert Bruckner, Beate List, and Josef Scheifer. Developing requirements for data warehouse systems with use cases. *AMCIS 2001 Proceedings*, page 66, 2001.

[23] Delvin R. Bunton. Wildland fire and weather information data warehouse. *UNITED STATES DEPARTMENT OF AGRICULTURE FOREST SERVICE GENERAL TECHNICAL REPORT NC*, pages 297–302, 2000.

[24] Liliana Cabral, Barry Norton, and John Domingue. The business process modelling ontology. In *Proceedings of the 4th international workshop on semantic business process management*, pages 9–16. ACM, 2009.

[25] Colin Camerer. *Behavioral game theory: experiments in strategic interaction*. The roundtable series in behavioral economics. Russell Sage Foundation ; Princeton University Press, New York, N.Y. : Princeton, N.J, 2003.

[26] G.A. Carpenter and S. Grossberg. The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88, March 1988.

[27] Gianfranco Chicco. Overview and performance assessment of the clustering methods for electrical load pattern grouping. *Energy*, 42(1):68–80, June 2012.

[28] Sungjin Cho, Jeon-Young Kang, Ansar-Ul-Haque Yasar, Luk Knapen, Tom Bellemans, Davy Janssens, Geert Wets, and

Chul-Sue Hwang. An Activity-based Carpooling Microsimulation Using Ontology. *Procedia Computer Science*, 19:48–55, January 2013.

[29] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. *DAML+OIL Reference Description*. World Wide Web Consortium, December 2001. http://www.w3.org/TR/daml+oil-reference.

[30] Université de Standford. Protege - éditeur d'ontologie.

[31] H. Dehainsala, G. Pierra, and L. Bellatreche. OntoDB: An Ontology-Based Database for Data Intensive Applications. In *Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07)*, volume 4443 of *Lecture Notes in Computer Science*, pages 497–508. Springer, 2007.

[32] Christelle Deschaseaux. Le Predit 4 s'achève, bientôt un Predit 5. *Energie Plus*, (516):11, January 2013.

[33] Sylvie Despres. Construction d'une ontologie modulaire pour l'univers de la cuisine numérique, 2014.

[34] Jérôme Euzenat, Christian Meilicke, Heiner Stuckenschmidt, Pavel Shvaiko, and Cássia Trojahn dos Santos. Ontology Alignment Evaluation Initiative: Six Years of Experience. *J. Data Semantics*, 15:158–192, 2011.

[35] Chimène Fankam, Ladjel Bellatreche, Dehainsala Hondjack, Yamine Ait Ameur, and Guy Pierra. SISRO, conception de bases de données à partir d'ontologies de domaine. *Technique et Science Informatiques*, 28(10):1233–1261, 2009.

[36] Chimène Fankam. *OntoDB2 : un systeme flexible et efficient de Base de Données à Base Ontologique pour le Web sémantique et les données techniques*. PhD thesis, ENSMA, Decembre 2009.

[37] Dieter Fensel, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, and Peter F. Patel-Schneider. OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

[38] Frederico Fonseca, Max Egenhofer, Peggy Agouris, and Gilberto Câmara. Using ontologies for integrated geographic information systems. *Transactions in GIS*, 6(3), 2002.

[39] Patrick Gagnol, Patrick Jochem, and W. Fichtner. CROME: the French and German Field Demonstration of the Interoperable Mobility with EVs. *Proceedings of EVS27*, 2013.

[40] María del Mar Roldán García, Ismael Navas Delgado, and José Francisco Aldana Montes. A Design Methodology for Semantic Web Database-Based Systems. In *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA'05)*, pages 233–237. IEEE Computer Society, 2005.

[41] Stephen P. Gardner. Ontologies and semantic data integration. *Drug Discovery Today*, 10(14):1001 – 1007, 2005.

[42] Gartner. Gartner Says More Than 50 Percent of Data Warehouse Projects Will Have Limited Acceptance or Will

Be Failures Through 2007. Technical report, February 2005.

[43] Francois Goasdoué, Véronique Lattès, and Marie-Christine Rousset. The use of carin language and algorithms for information integration: The picsel system. 2000.

[44] Cheng Hian Goh. *Representing and reasoning about semantic conflicts in heterogeneous information systems*. PhD thesis, MIT Sloan School of Management, 1997.

[45] Cheng Hian Goh, Stéphane Bressan, Stuart Madnick, and Michael Siegel. Context interchange: new features and formalisms for the intelligent integration of information. *ACM Transactions on Information Systems*, 17(3):270–293, July 1999.

[46] M. Golfarelli. Data warehouse lifecycle and design. In *Encyclopedia of Database Systems*, pages 658–664. Springer US, 2009.

[47] Matteo Golfarelli, Dario Maio, and Stefano Rizzi. The dimensional fact model: a conceptual model for data warehouses. *International Journal of Cooperative Information Systems*, 7(02n03):215–247, 1998.

[48] Matteo Golfarelli and Stefano Rizzi. *Data Warehouse Design: Modern Principles and Methodologies*. McGraw-Hill, Inc., 2009.

[49] Gouvernement. Une loi en faveur du développement des bornes de recharge pour véhicules électriques, July 2014.

[50] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, 1993.

[51] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies (IJHCS)*, 43(5-6):907–928, 1995.

[52] N. Guarino. Formal Ontology and Information Systems. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS'98)*, pages 3–15. IOS Press, 1998.

[53] M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel. Semantic business process management: a vision towards using semantic Web services for business process management. pages 535–540. IEEE, 2005.

[54] Dehainsala Hondjack, Guy Pierra, and Ladjel Bellatreche. Ontodb: An ontology-based database for data intensive applications. In *DASFAA*, pages 497–508, 2007.

[55] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From $\mathcalSHIQ$ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[56] William H. Inmon. *Building the data warehouse*. J. Wiley, New York, 3rd ed edition, 2002.

[57] Ivar Jacobson, Grady Booch, and James Rumbaugh. *The unified software development process*. Addison-Wesley, Reading, Mass, 1999.

[58] S. Jean, H. Dehainsala, D Nguyen Xuan, G. Pierra, L. Bel-

latreche, and Yamine Aït-Ameur. OntoDB: It is Time to Embed your Domain Ontology in your Database. In *Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07) (Demo Paper)*, pages 1119–1122, 2007.

[59] Stéphane Jean, Guy Pierra, and Yamine Aït-Ameur. OntoQL: an exploitation language for OBDBs. In *Proceedings of the VLDB 2005 PhD Workshop. Co-located with the 31th International Conference on Very Large Data Bases (VLDB'05)*, pages 41–45, 2005.

[60] Stéphane Jean, Guy Pierra, and Yamine Aït Ameur. Domain Ontologies: A Database-Oriented Analysis. In *Web Information Systems and Technologies, International Conferences, WEBIST 2005 and WEBIST 2006. Revised Selected Papers*, Lecture Notes in Business Information Processing, pages 238–254. Springer Berlin Heidelberg, 2007.

[61] Lihong Jiang, Hongming Cai, and Boyi Xu. A Domain Ontology Approach in the ETL Process of Data Warehousing. In *IEEE 7th International Conference on e-Business Engineering, ICEBE 2010, Shanghai, China, November 10-12, 2010*, pages 30–35, 2010.

[62] Jean-Pierre Kahane and Pierre Gilles Lemarié-Rieusset. *Séries de Fourier et ondelettes*. Cassini, Paris, 1998.

[63] Selma Khouri and Ladjel Bellatreche. DWOBS: Data Warehouse Design from Ontology-Based Sources. In JeffreyXu Yu, MyoungHo Kim, and Rainer Unland, editors, *Database Systems for Advanced Applications*, volume 6588 of *Lecture Notes in Computer Science*, pages 438–441. Springer Berlin Heidelberg, 2011.

[64] Ralph Kimball. *The data warehouse toolkit: practical techniques for building dimensional data warehouses*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[65] Ralph Kimball, editor. *The data warehouse lifecycle toolkit*. Wiley Pub, Indianapolis, IN, 2nd ed edition, 2008.

[66] Ryan KL Ko. A computer scientist's introductory guide to business process management (BPM). *Crossroads*, 15(4):4, 2009.

[67] James Kobielus. Data Scientists: Explore Game Theory to Boost Customer Engagement, August 2012.

[68] Pyrros Koletsis and Euripides GM Petrakis. SIA: semantic image annotation using ontologies and image content analysis. In *Image Analysis and Recognition*, pages 374–383. Springer, 2010.

[69] Tim Kraska and Uwe Röhm. Genea: Schema-aware mapping of ontologies into relational databases. In *COMAD*, pages 92–103. Tata McGraw-Hill Publishing Company Limited, 2006.

[70] Philippe Kruchten. From Waterfall to Iterative Development—A Challenging Transition for Project Managers. *Rational Edge, Rational Software*, 2001.

[71] Darl Kuhn, Sam R Alapati, and Bill Padfield. SQL Access Advisor. In *Expert Indexing in Oracle Database 11g*, pages 233–248. Springer, 2011.

[72] Bellatreche Ladjel. *Contributions à la Conception et l'Exploitation des Sys-*

*tèmes d'Intégration de Données*. PhD thesis, November 2009.

[73] Craig Larman and Victor R. Basili. Iterative and incremental development: A brief history. *Computer*, 36(6):47–56, 2003.

[74] LDC. Linked Data, 2014.

[75] Samuel Le Feon. *Evaluation environnementale des besoins de mobilité des grandes aires urbaines en France - Approche par Analyse de Cycle de Vie*. PhD thesis, 2014. Thèse de doctorat dirigée par Laforest, Valérie Sciences et Génie de l'Environnement Saint-Etienne, EMSE 2014 2014EMSE0729.

[76] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 233–246. ACM, 2002.

[77] Jing Lu, Li Ma, Lei Zhang, Jean-Sébastien Brunner, Chen Wang, Yue Pan, and Yong Yu. Sor: a practical system for ontology storage, reasoning and search. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 1402–1405. VLDB Endowment, 2007.

[78] Sean Luke. Ontology-based knowledge discovery on the world-wide web. In *Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence (AAAI96*, pages 96–102. AAAI Press, 1996.

[79] Li Ma, Zhong Su, Yue Pan, Li Zhang, and Tao Liu. Rstar: an rdf storage and query system for enterprise resource management. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 484–491, New York, NY, USA, 2004. ACM.

[80] Cristina Maier, Debabrata Dash, Ioannis Alagiannis, Anastasia Ailamaki, and Thomas Heinis. Parinda: an interactive physical designer for postgresql. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 701–704. ACM, 2010.

[81] Data and Knowledge Management. BPMN Ontology.

[82] Bery Leouro Mbaiossoum. *Conception physique des bases de données à base ontologique : le cas des vues matérialisées*. PhD thesis, December 2014.

[83] Eduardo Mena, Arantza Illarramendi, Vipul Kashyap, and AmitP. Sheth. OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.

[84] Roger B Myerson. *Game theory analysis of conflict*. Harvard University Press, Cambridge, Mass., 1991.

[85] Victoria Nebot and Rafael Berlanga. Building data warehouses with semantic data. In *Proceedings of the 2010 EDBT/ICDT Workshops*, page 9. ACM, 2010.

[86] Victoria Nebot and Rafael Berlanga. Building data warehouses with semantic web data. *Decision Support Systems*, 2011.

[87] Dung Nguyen-Xuan. *Intégration de base de données hétérogènes par articulation ıta priori d'ontologies : application aux catalogues de composants industriels*. PhD thesis, LISI/ENSMA et Université de Poitiers, 2006.

[88] Sree Nilakanta, Kevin Scheibe, and Anil Rai. Dimensional issues in agricultural data warehouse designs. *Computers and Electronics in Agriculture*, 60(2):263 – 278, 2008.

[89] S.L. Nimmagadda and H. Dreher. Ontology based data warehouse modeling and mining of earthquake data: prediction analysis along Eurasian-Australian continental plates. In *Proceedings of Industrial Informatics, 2007 5th IEEE International Conference*, volume 1, pages 597–602. IEEE, June 2007.

[90] Natalya F. Noy and Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, Stanford Knowledge Systems Laboratory, March 2001.

[91] Oracle. Oracle Semantic Technologies Overview.

[92] Zhengxiang Pan and Jeff Heflin. Dldb: Extending relational databases to support semantic web queries. In *In PSSS*, pages 109–113, 2003.

[93] Myung-Jae Park, Jihyun Lee, Chun-Hee Lee, Jiexi Lin, Olivier Serres, and Chin-Wan Chung. An efficient and scalable management of ontology. In *Advances in Databases: Concepts, Systems and Applications*, pages 975–980. Springer, 2007.

[94] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. *OWL Web Ontology Language Semantics and Abstract Syntax*. World Wide Web Consortium, February 2004. http://www.w3.org/TR/owl-semantics/.

[95] Carlos Pedrinaci, John Domingue, Christian Brelage, Tammo van Lessen, Dimka Karastoyanova, and Frank Leymann. Semantic Business Process Management: Scaling Up the Management of Business Processes. pages 546–553. IEEE, August 2008.

[96] G. Pierra, E. Sardet, J. C. Potier, G. Battier, J. C. Derouet, N. Willmann, and A. Mahir. Exchange of component data: the PLIB (ISO 13584) model, standard and tools. *Proceedings of the CALS EUROPE*, 98:160–176, 1998.

[97] Guy Pierra. Context-Explication in Conceptual Ontologies: The PLIB Approach. In R. Jardim-Gonçalves, J. Cha, and A. Steiger-Garçao, editors, *Proceedings of the 10th ISPE International Conference on Concurrent Engineering (CE 2003)*, pages 243–254, Madeira, Portugal, UNINOVA, A.A. Balkema, July 2003.

[98] Guy Pierra. Context Representation in Domain Ontologies and its Use for Semantic Integration of Data. *Journal Of Data Semantics (JODS)*, X:34–43, 2007.

[99] Guy Pierra, Hondjack Dehainsala, Yamine Aït-Ameur, and Ladjel Bellatreche. Base de Données à Base Ontologique : principes et mise en øeuvre. *Ingénierie des Systèmes d'Information*, 10(2):91–115, 2005.

[100] Chaussecourte Pierre, Glimm Birte, Horrocks Ian, Motik Boris, and Pierre Laurent. The Energy Management Adviser at EDF.

[101] Magali Pierre. Utiliser un véhicule hybride rechargeable en milieu professionnel : deux systèmes de prescriptions à l'origine de l'élaboration de cadres d'usages. July 2013.

[102] Magali Pierre, Christophe Jemelin, and Nicolas Louvet. Driving an electric vehicle. A sociological analysis on pioneer users. *Energy Efficiency*, 4(4):511–522, November 2011.

[103] La Poste and ERDF. Déjà un an d'expérimentation dans le cadre du projet Infini Drive. September 2013.

[104] Robert Powell. *Nuclear deterrence theory: The search for credibility*. Cambridge University Press, 1990.

[105] Valérie Psyché, Olvado Mendes, and Jacqueline Bourdeau. Apport de l'ingénierie ontologique aux environnements de formation à distance. *Revue Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation (STICEF)*, 10, 2003.

[106] Alan L. Rector, Matthew Horridge, and Nick Drummond. Building Modular Ontologies and Specifying Ontology Joining, Binding, Localizing and Programming Interfaces in Ontologies Implemented in OWL. In *AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering*, pages 69–73, 2008.

[107] Oscar Romero, Alkis Simitsis, and Alberto Abelló. Gem: requirement-driven generation of etl and multidimensional conceptual designs. *Data Warehousing and Knowledge Discovery*, pages 80–95, 2011.

[108] Catherine Roussey, Sylvie Calabretto, and Jean-Marie Pinon. Le thésaurus sémantique : contribution à l'ingénierie des connaissances documentaires. In *Actes des 6èmes Journées Ingénierie des Connaissances*, pages 209–220, 2002.

[109] Ségolène Royal. Plan de relance du logement : des mesures ambitieuses pour la cohésion sociale qui contribuent à la transition énergétique et la croissance verte, August 2014.

[110] Winston W. Royce. Managing the development of large software systems. In *proceedings of IEEE WESCON*, volume 26. Los Angeles, 1970.

[111] Kevin Royer, Ladjel Bellatreche, and Stéphane Jean. Combining domain and business ontologies in a modular construction method: EDF study case. In *Proceedings of the 38th International Convention, Business Intelligence Systems (MIPRO)*, pages 1452–1457, Opatija, Croatie, June 2014. IEEE.

[112] Kevin Royer, Ladjel Bellatreche, and Stéphane Jean. One semantic data warehouse fits both electrical vehicle data and their process. In *Proceedings of the 17th International IEEE Confrence on Intelligent Transportation Systems (ITSC2014)*, Qingdao, China, October 2014. IEEE.

[113] Kevin Royer, Ladjel Bellatreche, Anne Le-Mouel, and Gilbert Schmitt. Un entrepôt de données pour l'analyse de la recharge des véhicules électriques : un

retour d'expérience. pages 118–127, Bordeaux, June 2012. RNTI.

[114] Kevin Royer and François De Sousa. Etude de saisonnalité sur les données Kleber. Technical report, January 2014.

[115] Alkis Simitisis, Panos Vassiliadis, Spiros Skiadopoulos, and Timos Sellis. Data warehouse refreshment. 2007.

[116] Alkis Simitsis, Dimitrios Skoutas, and Malú Castellanos. Natural language reporting for etl processes. In *Proceedings of the ACM 11th international workshop on Data warehousing and OLAP*, pages 65–72. ACM, 2008.

[117] Alkis Simitsis, Dimitrios Skoutas, and Malú Castellanos. Representation of conceptual etl designs in natural language using semantic web technology. *Data & Knowledge Engineering*, 69(1):96–115, 2010.

[118] Alkis Simitsis and Panos Vassiliadis. A methodology for the conceptual modeling of etl processes. In *CAiSE workshops*, 2003.

[119] Dimitrios Skoutas and Alkis Simitsis. Designing etl processes using semantic web technologies. In *Data Warehousing and OLAP: Proceedings of the 9 th ACM international workshop on Data warehousing and OLAP*, volume 10, pages 67–74, 2006.

[120] Dimitrios Skoutas and Alkis Simitsis. Ontology-based conceptual design of etl processes for both structured and semi-structured data. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(4):1–24, 2007.

[121] Dimitrios Skoutas and Alkis Simitsis. Ontology-based conceptual design of ETL processes for both structured and semi-structured data. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(4):1–24, 2007.

[122] Michael K. Smith, Chris Welty, and Deborah L. McGuinness. *OWL Web Ontology Language Guide*. World Wide Web Consortium, February 2004. http://www.w3.org/TR/owl-guide/.

[123] Peter Spyns. Data modelling versus Ontology engineering. *SIGMOD Record*, 31:12–17, 2002.

[124] Olivier Steichen, Christel Daniel-Le Bozec, Marie-Christine Jaulent, Jean Charlet, and others. Construction d'une ontologie pour la prise en charge de l'hypertension artérielle. *18es Journées Francophones d'Ingénierie des Connaissances*, 2007.

[125] Lynn Andrea Stein, Dan Connolly, and Deborah L. McGuinness. *DAML-ONT Initial Release*. 2000. http://www.daml.org/2000/10/daml-ont.html.

[126] Thomas Stöhr, Robert Müller, and Erhard Rahm. An integrative and uniform model for metadata management in data warehousing environments. In *Proceedings of the International Workshop on Design and Management of Data Warehouses, Heidelberg, Germany*, volume 189, 1999.

[127] Heiner Stuckenschmidt and Holger Wache. Context Modeling and Transformation for Semantic Interoperability. In *In Knowledge Representation Meets Databases (KRDB*, 2000.

[128] R. Subhashin and J. Akilandeswari. A survey on ontology construction metho-

dologies. *International Journal of Enterprise Computing and Business System,(Online)*, 1(1), 2011.

[129] Vijayan Sugumaran and Veda C. Storey. The role of domain ontologies in database design: An ontology management and conceptual modeling environment. *ACM Transactions on Database Systems (TODS)*, 31(3):1064–1094, September 2006.

[130] Henry Valéry TEGUIAK, Guy PIERRA, Yamine AIT-AMEUR, and Ladjel BELLATRECHE. *Construction d'ontologies à partir de textes: une approche basée sur les transformations de modèles*. PhD thesis, December 2012.

[131] Tesla. Tesla S, 2014.

[132] Juan Trujillo and Sergio Luján-Mora. A uml based approach for modeling etl processes in data warehouses. *Conceptual Modeling - ER 2003*, pages 307–320, 2003.

[133] Bor-Yuan Tsai, Simon Stobart, Norman Parrington, and Barrie Thompson. Iterative Design and Testing within the Software Development Life Cycle. *Software Quality Journal*, pages 295–309, December 1997.

[134] M. Uschold and R. Jasper. A Framework for Understanding and Classifying Ontology Applications. In *Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm, Sweden, (August 1999).*, 1999.

[135] J. Vallet, O. Brun, and B. Prabhu. A Game-theoretic Algorithm for Nonlinear Single-Path Routing Problems. In *Proceedings of the 7th conference on International Network Optimization Conference*, 2015.

[136] Wil MP van der Aalst. Patterns and xpdl: A critical evaluation of the xml process definition language. *BPM Center Report BPM-03-09, BPMcenter. org*, 2003.

[137] Panos Vassiliadis, Alkis Simitsis, and Eftychia Baikousi. A taxonomy of etl activities. In *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP*, pages 25–32. ACM, 2009.

[138] Panos Vassiliadis, Alkis Simitsis, Panos Georgantas, Manolis Terrovitis, and Spiros Skiadopoulos. A generic and customizable framework for the design of etl scenarios. *Information Systems*, 30(7):492–525, 2005.

[139] Panos Vassiliadis, Alkis Simitsis, and Spiros Skiadopoulos. On the logical modeling of ETL processes. In *Advanced Information Systems Engineering*, pages 782–786. Springer, 2002.

[140] Pepijn R. S. Visser, Martin D. Beer, Trevor J. M. Bench-Capon, B. M. Diaz, and Michael J. R. Shave. Resolving Ontological Heterogeneity in the KRAFT Project. In *Proceedings of the 10th International Conference on Database and Expert Systems Applications (DEXA'99)*, pages 668–677, September 1999.

[141] Raphael Volz, Daniel Oberle, Steffen Staab, and Boris Motik. Kaon server - a semantic web management system. In *Alternate Track Proceedings of the Twelfth International World Wide Web Conference, WWW2003, Budapest, Hungary, 20-24 May 2003*. ACM, 2003.

[142] W3C. W3c RDF Schema, February 2014.

[143] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based Integration of Information — a Survey of Existing Approaches. In *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing*, pages 108–117, 2001.

[144] John Wang. *Encyclopedia of data warehousing and mining*. Information Science Reference, Hershey, 2009.

[145] Branimir Wetzstein, Zhilei Ma, Agata Filipowska, Monika Kaczmarek, Sami Bhiri, Silvestre Losada, Jose-Manuel Lopez-Cob, and Laurent Cicurel. Semantic Business Process Management: A Lifecycle Based Requirements Analysis. In *SBPM*, 2007.

[146] Kevin Wilkinson, Craig Sayers, Harumi Kuno, Dave Reynolds, et al. Efficient rdf storage and retrieval in jena2. In *Proceedings of SWDB*, volume 3, pages 7–8, 2003.

[147] Dung Nguyen Xuan, Ladjel Bellatreche, and Guy Pierra. OntoDaWa, un système d'intégration à base ontologique de sources de données autonomes et évolutives. *Ingénierie des Systèmes d'Information*, 13(2):97–125, 2009.

[148] Daniel C Zilio, Jun Rao, Sam Lightstone, Guy Lohman, Adam Storm, Christian Garcia-Arellano, and Scott Fadden. DB2 design advisor: integrated automatic physical database design. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1087–1097. VLDB Endowment, 2004.

# Table des figures

# Liste des tableaux

# Glossaire

**ADEME** : Agence De l'Environnement et de la Maîtrise de l'Énergie
**API** : Application Programming Interface

**B2B** : Business to Business
**BDBO** : Base de Données à Base Ontologique
**BDD** : Base de Données
**BMW** : Bayerische Motoren Werke
**BPMN** : Business Process Modelisation and Notation

**CIFRE** : Convention Industrielle de Formation par la REcherche
**CROME** : CROss-border Mobility for EV
**CRUD** : Create, Read, Update, Delete
**CSV** : Comma-Separated Values

**DAML** : DARPA Agent Markup Language
**DARPA** : Defense Advanced Research Projects Agency
**DME** : Direction de la Mobilité Électricité
**DSA** : Data Staging Area
**DTVE** : Direction Transports et Véhicules Électriques

**EC** : Entrepôt de Connaissances
**ECF** : Entrepôt de Connaissances Flottant
**ED** : Entrepôt de Données
**EDBO** : Entrepôt de Données à Base Ontologique
**EDF** : Électricité De France
**EDF R&D** : Électricité De France Recherche et Développement
**ELT** : Extract, Load and Transform
**ETL** : Extract, Transform and Load

**GPS** : Global Positionning System
**GRPS**: General Packet Radio Service
**GSM** : Global System for Mobile

**IBM** : International Business Machines
**IP** : Internet Protocol

**ME** : Mobilité Électrique
**MINERVE** : Moyens d'INformations dédiés aux ExpéRimentations de Véhicules Électriques

**LIAS** : Laboratoire d'Informatique et d'Automatique pour les Systèmes

**OIL** : Ontology Inference Layer
**OLAP** : OnLine Analytical Processing
**OWL** : Ontology Web Language

**PLIB** : Parts LIBrary
**PM** : Processus Métier
**PREDIT** : Programme de Recherche Et D'Innovation dans les Transports Terrestres

**RDF** : Resource Description Framework

**SAVE** : Seine Aval Véhicule Électrique
**SCADA** : Supervisory Control And Data Acquisition
**SI** : Système d'Informations
**SIA** : Semantic Image Annotation
**SISRO** : Spécialisation, Importation Sélective et Représentation des Ontologies
**SGBD** : Système de Gestion de Base de Données
**SQL** : Structured Query Language

**UML** : Unified Modeling Language
**URI** : Uniform Resource Identifier

**VE** : Véhicule Électrique
**VHR** : Véhicule Hybride Rechargeable

**W3C** : World Wide Web Consortium

**XML** : eXtensible Markup Language
**XPDL** : XML Process Definition Language

**Vers un entrepôt sémantique des données et des processus métiers : le cas de la mobilité électrique chez EDF**
**par Kevin ROYER**

**Résumé :**

Le marché du véhicule électrique ($\mathcal{VE}$) est aujourd'hui en plein essor et il s'agit d'un marché qui représente un intérêt pour des industriels comme EDF. Pour réaliser ses objectifs (optimisation de la consommation, tarification...) EDF doit d'abord extraire des données hétérogènes (issues des $\mathcal{VE}$ et des bornes de recharge) puis les analyser. Pour cela nous nous sommes orientés vers un entrepôt de données ($ED$) qui est ensuite exploité par les processus métiers ($PM$). Afin d'éviter le phénomène *Garbage In/Garbage Out*, les données doivent être traitées. Nous avons choisi d'utiliser une ontologie pour réduire l'hétérogénéité des sources de données. La construction d'une ontologie étant lente, nous avons proposé une solution incrémentale à base briques ontologiques modulaires liées entre elles. La construction de l'$ED$, basé sur l'ontologie, est alors incrémentale. Afin de charger des données dans l'$ED$, nous avons défini les processus $ETL$ (*Extract, Transform & Load*) au niveau sémantique. Ensuite nous avons modélisé les $PM$ répétitifs selon les spécifications $BPMN$ (*Business Process Modelisation & Notation*) pour extraire les connaissances requises par EDF de l'$ED$. L'$ED$ constitué possède les données et des $PM$, le tout dans un cadre sémantique. Nous avons implémenté cela sur la plateforme *OntoDB* développée au Laboratoire d'Informatique et d'Automatique pour les Systèmes de l'ISAE-ENSMA. Elle nous a permis de manipuler l'ontologie, les données et les $PM$ d'une manière homogène grâce au langage *OntoQL*. De plus, nous lui avons fourni la capacité d'exécuter automatiquement les $PM$. Cela nous a permis de fournir à EDF une plate-forme adaptée à leurs besoins à base d'éléments déclaratifs.

**Mots-clés :** Véhicules électriques, Ontologies (informatique), Entrepôts de données, Exploration de données, Sémantique selon la théorie des jeux–Applications industrielles

**Abstract :**

Nowadays, the electrical vehicles ($\mathcal{EV}$) market is undergoing a rapid expansion and has become of great importance for utility companies such as EDF. In order to fulfill its objectives (demand optimization, pricing, etc.), EDF has to extract and analyze heterogeneous data from $\mathcal{EV}$ and charging spots. In order to tackle this, we used data warehousing ($DW$) technology serving as a basis for business process ($BP$). To avoid the garbage in/garbage out phenomena, data had to be formatted and standardized. We have chosen to rely on an ontology in order to deal with data sources heterogeneity. Because the construction of an ontology can be a slow process, we proposed an modular and incremental construction of the ontology based on bricks. We based our $DW$ on the ontology which makes its construction also an incremental process. To upload data to this particular $DW$, we defined the $ETL$ (*Extract, Trasform & Load*) process at the semantic level. We then designed recurrent $BP$ with $BPMN$ (*Business Process Modelization & Notation*) specifications to extract EDF required knowledge. The assembled $DW$ possesses data and $BP$ that are both described in a semantic context. We implemented our solution on the *OntoDB* platform, developed at the ISAE-ENSMA Laboratory of Computer Science and Automatic Control for Systems. The solution has allowed us to homogeneously manipulate the ontology, the data and the $BP$ through the *OntoQL* language. Furthermore, we added to the proposed platform the capacity to automatically execute any $BP$ described with $BPMN$. Ultimately, we were able to provide EDF with a tailor made platform based on declarative elements adapted to their needs.

**Keywords :** Electric vehicles, Ontologies (Information retrieval),Data warehousing, Data mining, Game-theoretical semantics–Industrial applications