MANAGEMENT SYSTÈMES D'INFORMATION

MDM

Enjeux et méthodes de la gestion des données

logica management consulting

Franck Régnier-Pécastaing
Michel Gabassi
Jacques Finet

DUNOD

MDM

Enjeux et méthodes de la gestion des données

Franck Régnier-Pécastaing

Responsable des offres Entreprise Information & Master Data Management chez Logica Management Consulting

Michel Gabassi

Ingénieur architecte à la Direction Informatique et Télécommunications d'EDF/GDF Suez

Jacques Finet

Ingénieur urbaniste à la Direction Informatique et Télécommunications d'EDF/GDF Suez

logica management consulting

DUNOD

Illustration de couverture : © Jean-Michel Pouget - Fotolia.com

ISBN 978-2-10-053555-2

Table des matières

Annexes

Avant-propos

Objectifs de l'ouvrage

La gestion des données est un sujet souvent négligé par les entreprises. En effet, les applications qui gèrent leurs activités implémentent avant tout des processus, les données n'étant qu'un corollaire parfois sous-évalué et sous-investi.

Pourtant, les données sont potentiellement une source de valeur, mais pour cela il est indispensable de s'attacher à **améliorer leur qualité et à limiter les incohérences** entre processus et applications.

Parmi ces données, certaines sont plus essentielles que d'autres car utilisées dans plusieurs applications : ce sont **les données de référence** ou « ***master data*** ».

L'objectif de ce livre est d'exposer les méthodes et solutions pour mieux gérer les données, et plus particulièrement les données de référence car elles structurent les applications du système d'information. Nous aborderons en particulier la notion essentielle de « **point de vérité** », sans laquelle les risques d'incohérence et de non-qualité sont importants.

À qui s'adresse ce livre ?

Ce livre s'adresse aussi bien aux **maîtrises d'ouvrage** soucieuses de mieux maîtriser la gestion des données de l'entreprise qu'aux **maîtrises d'œuvre** souhaitant analyser les méthodes et solutions.

Il est donc destiné à tous les responsables qui cherchent à améliorer la valeur de l'information détenue et utilisée par l'entreprise et souhaitent comprendre les enjeux, besoins, technologies, produits et organisations liés à la gestion des données.

Dirigeants, responsables métier, urbanistes, chefs de projet, etc. y trouveront les indications utiles pour transformer une vision stratégique liée aux données en une réalité, que cette vision soit métier (convergence orientée client, par exemple) ou système d'information (architecture orientée services, par exemple).

Comment lire l'ouvrage ?

Ce livre se décline comme suit :

- La première partie expose les **concepts, besoins et enjeux**. Il s'agit de définir le vocabulaire et de montrer la valorisation possible des données. En effet, comme souvent en informatique, il est essentiel de bien cerner les notions utilisées (que l'on rappellera également dans un glossaire très complet).
- La deuxième partie présente les **technologies, les bonnes pratiques, les architectures et les solutions** pour améliorer la gestion des données. Dans ce contexte, l'intérêt porté aux logiciels de type référentiel MDM (*Master Data Management*) ainsi qu'aux outils de DQM (*Data Quality Management*) répond à la maturité grandissante de ces offres.
- La dernière partie propose des **méthodes et des organisations**, avec le concept-clé de **gouvernance des données**. Chaque étape importante est décrite, depuis l'identification des propriétaires de données, jusqu'à la constitution d'une organisation en charge du pilotage des référentiels.

Remerciements

Les auteurs remercient leurs hiérarchies (Logica et EDF) de leur avoir permis de rédiger et publier cet ouvrage.

Nous remercions en particulier :

Georges Abou Harb – Line Horizontal Services SOA International Logica – pour le soutien aux différentes initiatives de ses « ouailles » et pour les opportunités qu'il leur offre, notamment permettre l'écriture et la parution de ce livre.

Michel Giraud, chef du département Architecture & Solutions à la Direction Informatique et Télécommunications d'EDF, pour son accord bienveillant à la rédaction de cet ouvrage.

Jean-Pascal Boutier et Vincent Colombani pour leur soutien sans faille et leur compréhension.

François Rivard pour sa lecture et sa plume, son structuralisme Levis Straussien et sa présence d'un bout à l'autre de la rédaction de cet ouvrage.

Bruno Ryckman pour les emprunts que nous lui avons fait au sein de ce livre. Pour son soutien ainsi que son jovial sérieux.

Pierre Verger pour sa lecture et sa plume, pour son œil d'urbaniste éclairé qui voit où se loge la pertinence et qui sait déroger à la doctrine.

Olivier Mathurin, Matthieu Scholler et Olivier Lallement pour leur participation, leur aide à la rédaction de certaines parties de ce livre. Pour leur sérieux et leur implication dans le bien étrange univers du MDM. In Data Veritas…

Sébastien Delayre ou Monsieur SOA, Philippe Meret ou Monsieur BPM, Nicolas Bo ou Monsieur Distribution, Luc Geoffray ou Monsieur SAP.

Eric Marcou, qui, à force de challenger chaque idée, fait progresser les gens qui le côtoient.

Christophe Barriolade, Pierre Bonnet, Sylvie L'ollivier… et tous les autres.

Introduction

Le volume des données ne cesse de croître dans les entreprises et leur bonne gestion devient un problème crucial. En effet, chaque métier (marketing, vente, gestion des ressources humaines, comptabilité, production…) génère, et parfois gère, ses propres données. Dans la majorité des cas, elles sont dispersées en différents îlots, ce qui rend difficile la mise à disposition d'un système efficace de gestion (plus de 90 % des entreprises sont concernées selon une étude de *01 DSI* réalisée en France en 2007).

Notre objet n'est pas de nous arrêter à tel ou tel type de donnée spécifique, mais plutôt de proposer une vision, des outils et une démarche propres à s'adapter à l'ensemble des données à traiter.

Bien entendu les données sont une constante au sein des systèmes d'information (SI) et nous n'avons pas la prétention de réécrire 60 ans d'histoire de l'informatique. Cependant, plusieurs éléments nous encouragent **à établir un état des lieux et à proposer une approche synthétique d'une problématique transversale** :

- l'apparition d'applications capables de traiter spécifiquement de la gestion des données ;
- les nouvelles orientations technologiques au sein des systèmes d'information (SOA, BPM…) ;
- la valorisation et la protection de la donnée au service de l'entreprise ;
- la nécessité de se conformer aux réglementations de plus en plus nombreuses (CNIL, Art. 29, SOX, Bâle II…) ;
- l'émergence des notions de gouvernance des données associées à l'ensemble de ces mouvements.

Historiquement, le système d'information d'une entreprise s'est structuré autour des applications opérationnelles et des applications décisionnelles. La figure Intro.1 illustre cette situation. On trouve dans le système opérationnel de production toutes les applications transactionnelles de l'entreprise, que ce soit en *front*, *back-office* ou support (voir au chapitre 1 ces notions). Toutes ces applications génèrent la grande masse des données qui sont ensuite utilisées, notamment dans les applications décisionnelles, lesquelles ont besoin de données fiables et à jour pour produire les rapports et les analyses indispensables au pilotage de l'entreprise.

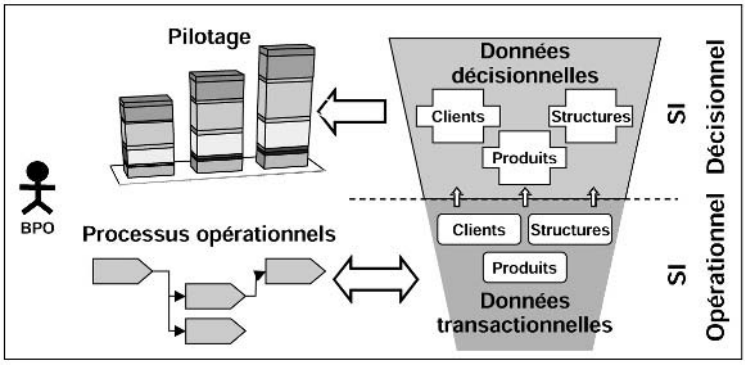

Figure Intro.1 – Système d'Information, modèle traditionnel

Dans cette vision traditionnelle, seuls les processus opérationnels métier et le pilotage associé ont une réelle valeur pour l'entreprise. **Les données sont dispersées dans les différentes applications** et il est souvent bien difficile de déterminer quelle application constitue la référence pour telle ou telle donnée métier. De même les problèmes d'incohérence entre applications, voire parfois entre sites d'hébergement d'une même application, sont difficilement maîtrisés.

Parmi toutes les données, certaines sont plus critiques pour l'activité métier et le système d'information car elles sont structurantes et largement partagées entre plusieurs applications : nous les appellerons **les données de référence**. Elles sont souvent disséminées, sans recherche de fertilisation croisée entre les différents processus et activités. Cela génère éventuellement des problèmes d'homogénéité et de rapprochements (dans les métiers et/ou dans le système d'information) et pénalise les gisements de valeurs liés à la donnée.

Prenons, comme exemple a priori fictif (voir figure Intro.2), le cas d'une entreprise qui utilise plusieurs canaux de communication avec ses clients : serveur web pour une application de vente en ligne, courrier électronique pour l'envoi des factures, téléphone pour la gestion des contacts de type demandes et réclamations. Si ces trois applications offrent leur propre vision client sans souci de partager une même référence, le client constatera vite l'incohérence du système d'information de cette entreprise. Il sera probablement perplexe, irrité, et doutera sérieusement de la capacité de la dite entreprise à gérer correctement ses clients !

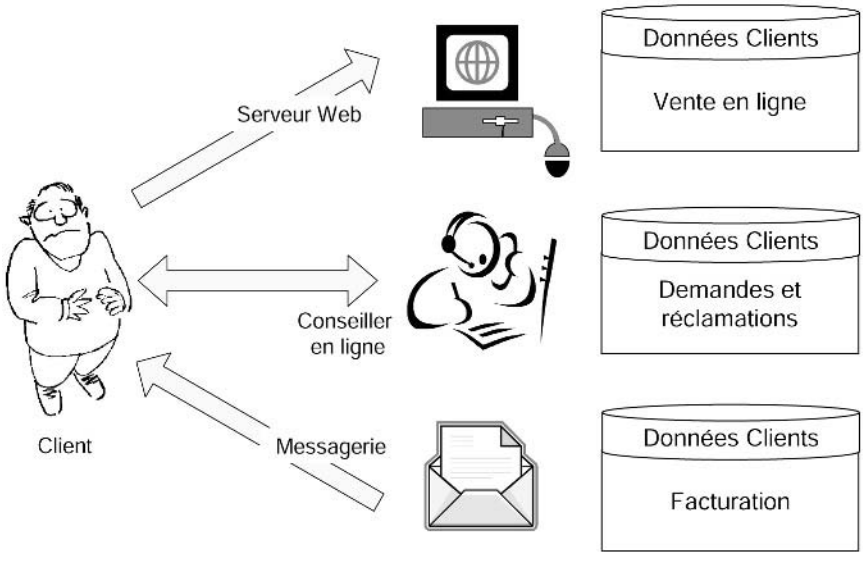

Figure Intro.2 – Un exemple de mauvaise gestion de données de référence

Il est ainsi évident que la notion de **MDM (*Master Data Management*)**, à savoir les disciplines nécessaires à « **la gestion des données de référence métier** », nécessite une rationalisation d'approches bien souvent disparates et hétérogènes. **Il s'agit de dresser le bilan des dernières évolutions dans la prise en compte et le traitement des données de référence par les système d'information bien sûr, mais peut-être encore plus par les métiers de l'entreprise.**

C'est ce que nous allons présenter dans cet ouvrage, qui se décline comme suit :

- La première partie de l'ouvrage s'attache à la **compréhension des notions**. Il s'agit en particulier de définir le vocabulaire et de montrer la valorisation possible des données.
- La deuxième partie présente **la mise en œuvre d'architectures et de produits** pour améliorer la gestion des données. Le MDM (*Master Data Management*) y est en particulier détaillé.
- La dernière partie propose des **méthodes et des organisations pour le pilotage**, avec le concept-clé de gouvernance des données.

PREMIÈRE PARTIE

Comprendre : les concepts

1

La gestion des données de référence

Objectif

Ce chapitre vise à répondre aux questions fondamentales relatives aux données de référence. Qu'est-ce qu'une donnée de référence ? Quels en sont les différents types ? Quels liens entretiennent-elles avec les autres notions relatives aux données ?

Il s'attache également aux questions relatives à leur gestion. Pourquoi constituent-elles un domaine de gestion à part entière ? Quels sont les besoins et les enjeux qui s'y rattachent, tant métier que **système d'information** ?

Il s'agit donc ici de fixer le vocabulaire de la gestion des données, de présenter les problèmes potentiels et d'évoquer les premières améliorations possibles.

1.1 PRINCIPES ET NOTIONS ÉLÉMENTAIRES

Dans les technologies de l'information, une **donnée** est une description élémentaire, souvent codée, d'un objet, d'une transaction d'affaire, d'un événement…

Les données ont une importance fondamentale pour toutes les activités de l'entreprise. Ces activités ou fonctions métier peuvent être de différents types : cœur de métier, *front*, *back* ou encore support. Ce sont, par exemple :

- La vente, qui est une activité *front* reposant sur les données produits et clients.

- La facturation en *back* et la comptabilité d'entreprise en *support* qui s'appuient sur des données telles que les prix et les comptes.
- Le pilotage de l'ensemble qui effectue la consolidation de toutes les données mobilisées lors des transactions.

La figure 1.1 schématise ces activités.

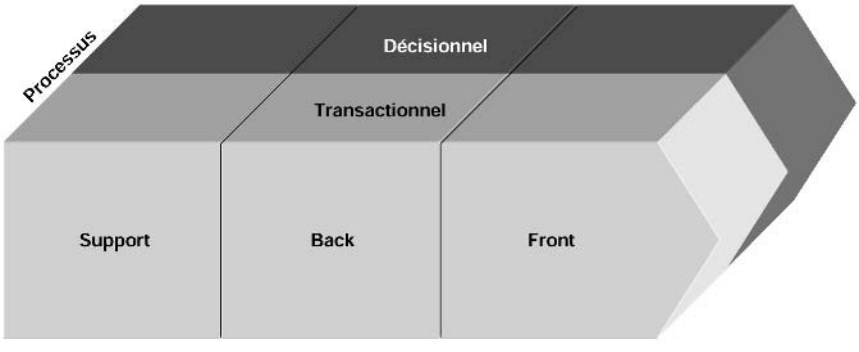

Figure 1.1 — Structure des activités de l'entreprise

Les activités de l'entreprise génèrent donc des données métier. Tous les processus métier utilisent des données comme éléments de base. Parmi ces données, certaines sont particulières car elles sont utilisées par :

- un grand nombre de processus métier ;
- plusieurs entités organisationnelles ;
- l'entreprise mais aussi ses partenaires.

Ces données sont généralement les **données de référence**. **La première caractéristique de ces données est leur partage, leur utilisation par plusieurs applications du système d'information**. Client, fournisseur, employé, article, contrat, offre sont des données de référence. Ces données sont les éléments de base de la gestion de toute entreprise. Elles sont indispensables à toute transaction commerciale, à tout contrat d'achat, à toute commande d'approvisionnement.

Dans la suite de l'ouvrage on utilisera les termes suivants que l'on considérera équivalents : donnée de référence, donnée maître, *master data*.

Tout ce que nous évoquerons à propos des données dites de référence peut être en général étendu à toute donnée partagée entre plusieurs processus et/ou applications.

La figure 1.2 présente quelques données de référence dans un système d'information. On voit que ces données, comme d'ailleurs l'ensemble des données du système d'information, sont dispersées et souvent dupliquées entre différents processus et applications.

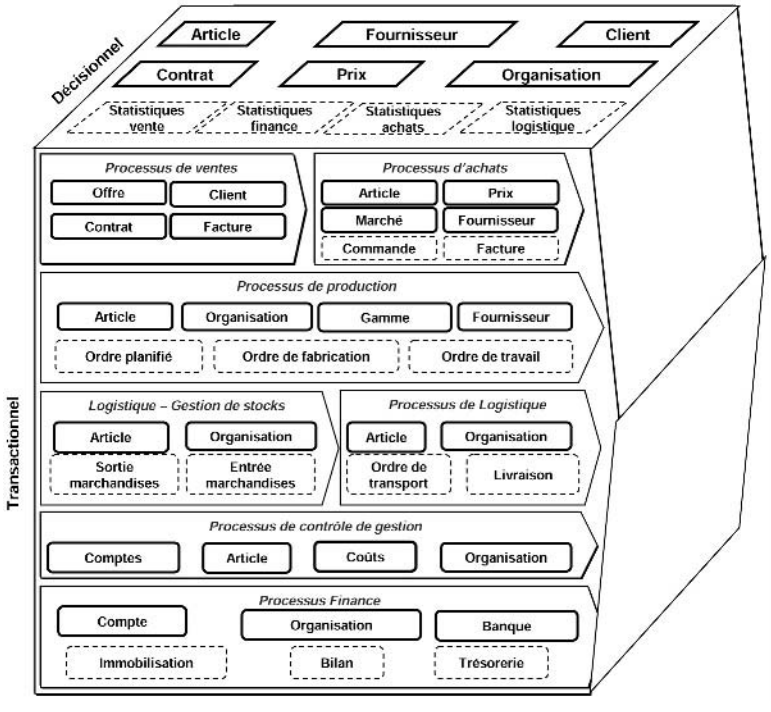

Figure 1.2 — Quelques données de référence dans un système d'information

1.1.1 Caractéristiques des données de référence

Selon que l'on se situe côté métier ou côté système d'information, différentes définitions peuvent être attribuées aux données de référence :

- Pour le manager, une donnée de référence est une information structurante de l'activité. Sa maîtrise offre un avantage à l'entreprise par une meilleure connaissance de son environnement et permet aux différents acteurs (internes ou partenaires) de travailler sur des bases communes.
- Pour le responsable de projet métier, une donnée de référence est une donnée des processus métier, partagée par plusieurs d'entre eux. Cette donnée revêt différentes dimensions en fonction de l'activité supportée par tel ou tel processus.
- Pour le responsable système d'information, une donnée de référence est une donnée **du système d'information, partagée et utilisée sans modifications**

par ses applications. Cette dernière caractéristique est essentielle pour identifier une donnée de référence.

La création ou la modification d'une donnée de référence est réalisée au travers de processus spécifiques. On nommera « processus référentiel métier » et par abus de langage « **processus référentiel** » les processus spécifiques de création ou de modification d'une donnée de référence. On nommera « processus opérationnel métier » ou « **processus métier** », les processus consommateurs structurés par les donnés de référence (voir chapitre 3 pour plus de détails sur ces notions).

Par essence, les données de référence ont, en général, **une relative stabilité** (relativement aux processus qui les utilisent). Une donnée de référence vit au-delà des instances de processus qui la mettent en œuvre. Par exemple, un client peut passer plusieurs commandes.

Les caractéristiques des données de référence peuvent évoluer sans remettre en cause leur nature ou leur « **unicité** ». Par exemple, un client peut déménager et donc changer d'adresse, mais il sera toujours identifié de la même façon.

Enfin, la donnée de référence doit être de **qualité** et respecter des **règles** communes à l'ensemble des entités utilisatrices.

1.1.2 Typologie des données de référence

On distingue trois grands types de données de référence qui appellent différents types de gouvernance et de socle technique (voir dans la suite ce que l'on entend par ces termes) :

- Les données « **maître** » sont en général les **objets métier principaux (« cœur de métier ») d'un domaine fonctionnel**. Ces données sont donc au cœur du **système d'information** et structurent les principales applications. En général, elles sont donc référencées dans de nombreuses applications.
  Exemples : client, article, fournisseur…
- Les données « **constitutives** » sont des **données constituées elles-mêmes d'attributs, qui caractérisent en général des données maître**, mais aussi d'autres objets métier.
  Exemple : adresse. Elle peut caractériser des données maître comme client, fournisseur… mais aussi d'autres objets métier comme interlocuteur (adresse postale) ou bon de livraison (adresse de livraison).
- Les données « **paramètre** » sont des **tables de valeurs** ou des **nomenclatures**.
  Exemples : codes postaux, codes devises, taux des taxes des communes. Par essence, ce sont les données les plus partagées au sein du **système d'information** et donc celles qui doivent faire l'objet d'une attention particulière.

La figure 1.3 montre un exemple de ces différents types de données de référence et des liens associés.

On notera que la nuance entre donnée « maître » et « constitutive » dépend du sujet de l'analyse. En effet, une donnée maître dans un certain périmètre peut s'avérer être constitutive au sein d'une vision plus large.

Les données de référence peuvent être, par ailleurs, communes à divers domaines fonctionnels ou spécifiques à un domaine fonctionnel particulier.

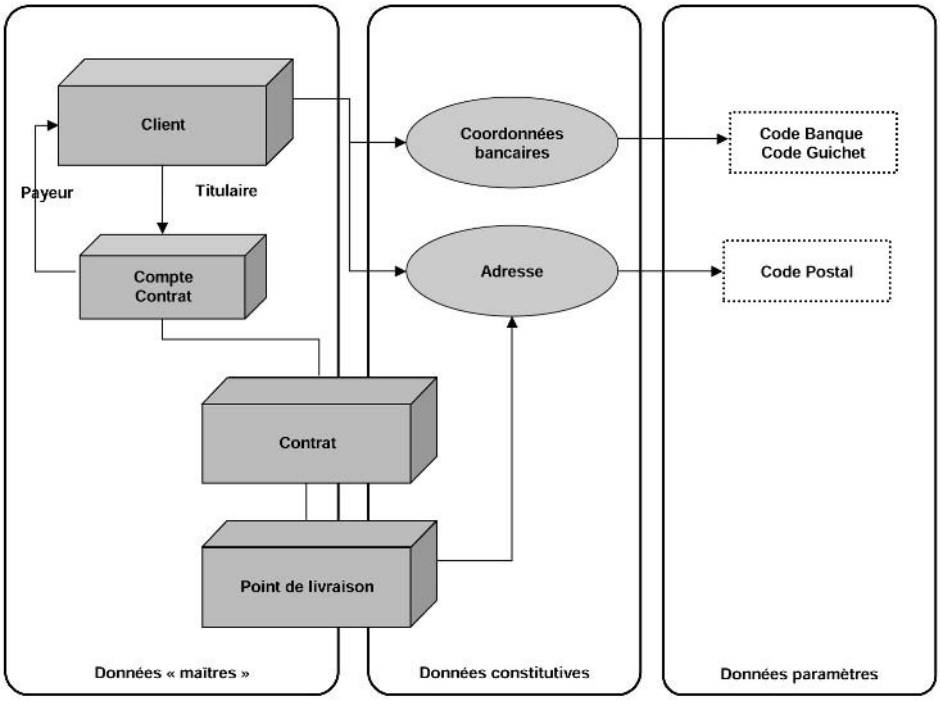

Figure 1.3 — Un exemple de différents types de données de référence.

Notons que cette distinction entre données « maître », « constitutive » et « paramètre » permet de mieux comprendre les différentes présentations de la suite de l'ouvrage mais n'est pas indispensable dans une étude informatique ou un projet.

1.1.3 La notion de famille de données

Si les différentes données de référence semblent pouvoir être analysées isolément, elles sont en fait liées entre elles, ce qui nous permet de parler de « famille de données ».

Ainsi, un référentiel « article » est lié à « fournisseur », un référentiel « client » est associé à « contrat » et pourtant les référentiels ne se substituent pas les uns aux autres.

On peut donc constituer des familles de données qui seront principalement dévolues à une activité de l'entreprise. Ceci aura notamment une incidence sur nos préconisations d'organisation et de gouvernance des données par la suite.

La figure 1.4 indique quelques exemples de données de référence et de regroupements propres à la structure d'une société de type « holding ».

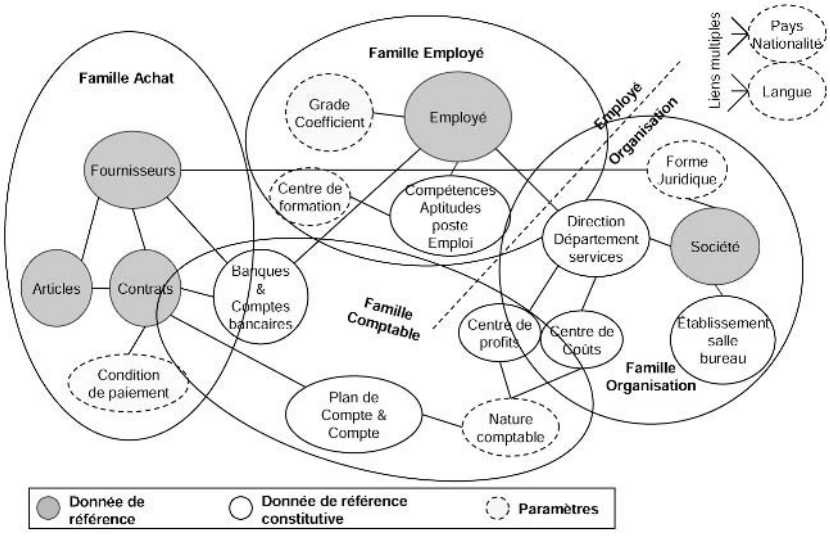

Figure 1.4 — Quelques exemples de regroupement en famille

1.1.4 Positionnement par rapport à d'autres notions

La figure 1.5 positionne les principaux concepts liés à la gestion des données de référence [1]. Ces concepts sont ensuite décrits en détail.

1. Voir le glossaire en fin d'ouvrage pour une définition complète de ces notions.

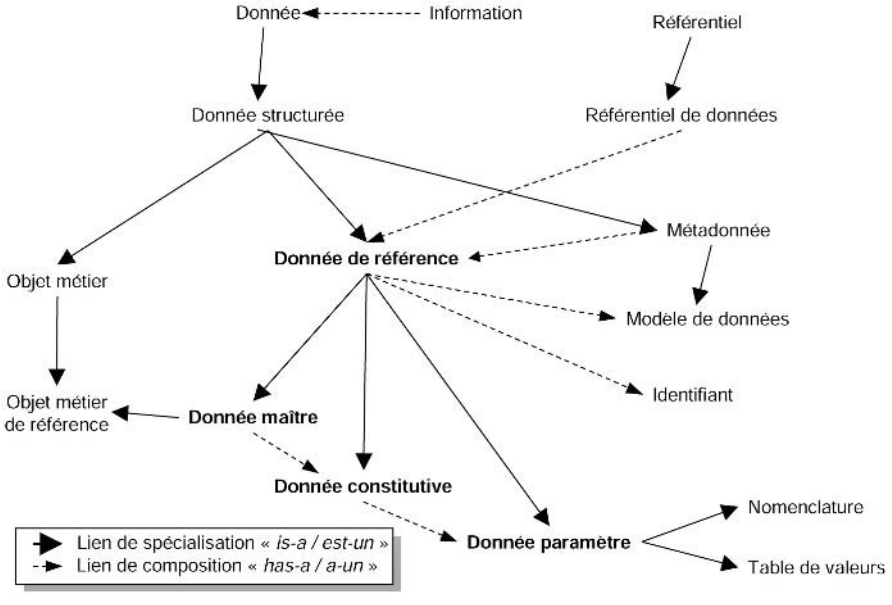

Figure 1.5 — Ontologie des données de référence

Donnée

Description élémentaire de nature numérique ou alphanumérique, représentée sous forme codée en vue d'être enregistrée, traitée, conservée et communiquée. Considérée individuellement, elle ne présente que peu d'intérêt humain et n'est donc utile que pour la machine.

Donnée structurée

Donnée dont on a établi fonctionnellement le sens de manière détaillée ainsi que les règles de création (dont les valeurs possibles dans le cas de listes) et enfin le moyen technique de représentation.

→ **Une donnée de référence est une catégorie de donnée structurée**.

Information

Données agrégées en vue d'une utilisation par l'homme (par exemple, le résultat d'une requête décisionnelle qui somme des données individuelles est une information). On parle aussi d'élément de connaissance susceptible d'être représenté à l'aide de conventions pour être conservé, traité ou communiqué (image, texte, donnée structurée).

La donnée sert à constituer l'information, cette dernière étant elle-même un élément de la connaissance (figure 1.6).

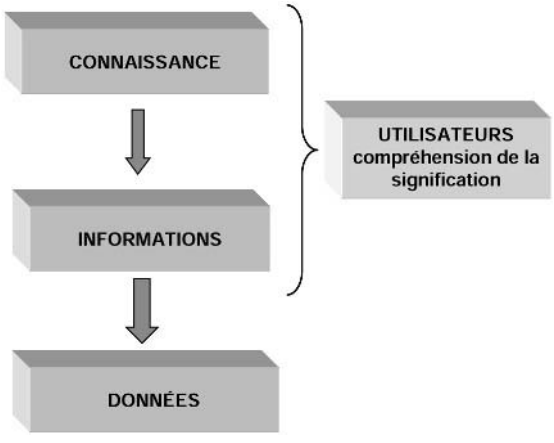

Figure 1.6 — Données, informations et connaissance

Objet métier

Ensemble d'informations homogènes du point de vue métier, manipulées dans le cadre des processus. C'est ce sur quoi porte une activité : l'objet métier matérialise le résultat du travail effectué par un acteur de l'entreprise. Ce sont des structures de données conçues pour représenter les livrables et les connaissances manipulées et gérées par les processus métier. Une structure de données est un ensemble organisé de données ayant quelque chose en commun et que l'on a groupées pour leur traitement (par exemple, la structure Identité est composée de civilité, nom, prénom, âge, profession).

→ Un objet métier peut être un objet de référence (donnée maître) s'il est structurant et partagé. Un objet métier peut lui-même inclure des données de référence (constitutive, paramètre).

Métadonnée

Littéralement, une donnée qui décrit une donnée. Nous reviendrons plus en détail sur cette notion dans les chapitres qui suivent.

→ Une métadonnée est une description métier et/ou technique d'une donnée.

Modèle de donnée

Représentation de l'ensemble des métadonnées ainsi que de leurs structures de regroupement et de relations sémantiques.

On distingue le modèle conceptuel du modèle logique et physique associé à un système de gestion de base de données (tables, attributs, clés). On y associera aussi la notion de modèle de flux (cf. annexe « Modélisation des données »).

Un modèle de données est donc une collection des descriptions des structures de données et des champs (attributs).

→ Dans le cadre de notre sujet, le modèle de données va servir à décrire précisément les données de référence, et en particulier leurs contextes d'utilisation avec l'ensemble de leurs données et relations.

Identifiant

Une donnée spécifique qui repère de manière unique un objet ou une information dans un système.

→ Un identifiant sert à repérer aussi les données de référence.

Exemple : le numéro SIRET identifie les établissements d'une entreprise.

Référentiel

Un référentiel s'apparente à un ensemble d'éléments dans lequel l'entreprise documente et agrège des règles de fonctionnement, techniques ou fonctionnelles.

Référentiel de données

Un référentiel de données est un « répertoire » clairement identifié (qui « fait référence »), qui stocke et permet la gestion et l'utilisation de données dans plusieurs traitements.

→ Un référentiel de données contient toutes sortes de données, de référence ou pas.

> Par abus de langage, on confond souvent gestion des données de référence, référentiel et référentiel de données.

Table de valeurs (ou domaine de valeurs)

Liste des valeurs possibles d'une donnée.

Exemples : liste de segments marketing.

Nomenclature

Table de valeurs codifiées d'une donnée. Il s'agit d'une donnée de référence de type paramètre.

Exemple : la donnée Activité principale de l'entreprise (ou code APE) est exprimée selon la Nomenclature des activités françaises (NAF).

1.1.5 Gestion des données de référence (GDR)

Nous introduisons la notion de gestion de données de référence ou GDR (ou encore MDM pour *Master Data Management*). Il s'agit uniquement ici de positionner ce concept par rapport à d'autres plus connus. Nous y reviendrons bien entendu très largement tout au long de l'ouvrage.

Gestion des données de référence (GDR) : contexte général

La GDR (gestion des données de référence, *Reference Data Management*), ou MDM (*Master Data Management*), est une discipline des technologies de l'information qui concerne les données de référence. Elle est le garant de la cohérence entre diverses architectures de systèmes et fonctions métier.

La GDR ou MDM inclut d'autres disciplines afin d'atteindre ses objectifs.

Ainsi le DQM (*Data Quality Management*) est une des technologies nécessaires aux traitements relatifs à la qualité des données. Par extension, car la GDR repose sur l'ensemble des couches d'infrastructure d'échange et d'intermédiation du **système d'information**, on pourra aussi inclure dans le champ d'étude les notions d'EAI (*Enterprise Application Integration*), ETL (*Extract Transform Load*), EII (*Enterprise Information Integration*) et ESB (*Enterprise Service Bus*), etc.

En outre, on considère généralement la GDR ou MDM comme une discipline nécessaire à la mise en œuvre d'une SOA (*Service Oriented Architecture*). En effet, une vision métier de la SOA, et en particulier la mise en œuvre de processus transverses via le BPM (*Business Process Management*), repose sur la fourniture de services de données.

Enfin, la GDR ou MDM n'est-elle pas une sous-division d'une discipline plus large qui décline les bonnes pratiques et outille l'ensemble des aspects « données et information » au sein du **système d'information** ? Cette discipline, qui se nomme EIM (*Enterprise Information Management*), reste cependant à ce jour plus un trigramme de cabinet d'analyse qu'une fonction d'entreprise.

La figure 1.7 présente cette typologie.

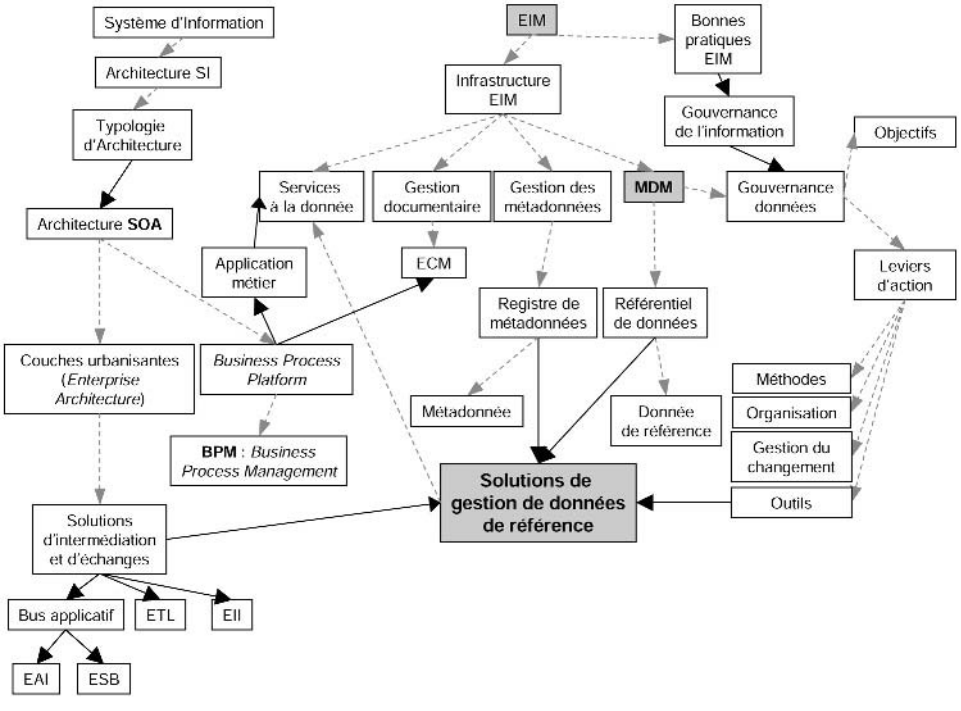

Figure 1.7 — Positionnement des termes

Les grandes entreprises et administrations disposent en général de systèmes informatiques qui outillent l'ensemble de leurs activités ou fonctions métier (recherche et développement, marketing, ventes, budget, finances…) et qui s'étendent au-delà des frontières, notamment pour les grands groupes. Ces différents systèmes ont souvent besoin de partager des données clés indispensables à la société mère (par exemple, les produits, les clients et fournisseurs). La problématique est la même quand deux entreprises ou plus veulent partager des données au-delà de leur propre périmètre. **La cohérence des données partagées entre différents systèmes informatiques est un besoin critique pour ces entreprises.**

Ainsi, la multiplicité des métiers, des pays, des lignes de produits, de services, des canaux de distribution, etc., contribuant à segmenter les activités de l'entreprise, génère une dissémination des référentiels (voir figure 1.8 l'exemple des référentiels marchands dans la grande distribution).

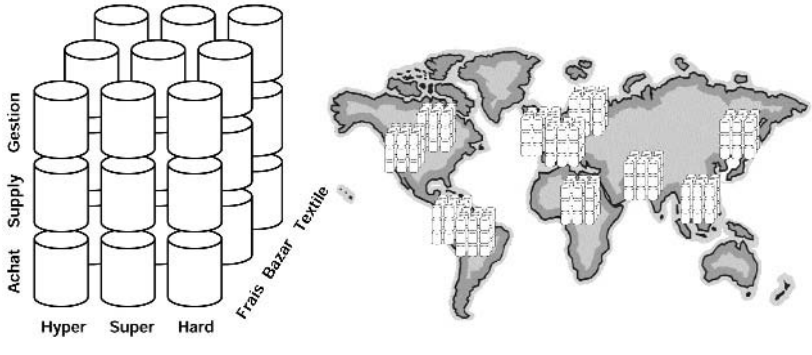

Figure 1.8 — La dissémination des référentiels marchands dans la distribution

Dans un modèle classique de système d'information, la gestion des données de référence est nécessaire pour coordonner entre eux les différents systèmes ERP, les systèmes légataires, les autres progiciels et le décisionnel qui supportent cette dissémination. La gestion des données est donc une discipline essentielle pour l'agilité et la cohérence des métiers (fusion, acquisition, international, fondement des processus) et du **système d'information**.

1.1.6 Enjeux et besoins

On distingue ici les enjeux (ce qui va permettre de réaliser des gains) des besoins (ce qui est nécessaire pour satisfaire un manque).

Les enjeux et besoins listés ci-dessous sont génériques et à resituer dans le contexte particulier de chaque entreprise. Nous verrons plus loin comment le cadre de gouvernance des données peut aider chacune d'entre elles à identifier ses propres enjeux et besoins.

Enjeux

Au niveau métier, on notera principalement :

- **Valoriser les données** (et en particulier les données de référence) comme actifs productifs (1,5 milliard de téra-octets sont créés tous les ans dans le monde).
- Disposer de **données fiables et à jour**.
- **Réduire le temps des cycles métier** liés aux données de référence ou aux processus référentiels.
- Améliorer et fiabiliser les processus grâce à des **données de référence accessibles et de qualité**.
- Gouverner ou piloter les données, comme cela est déjà fait pour les processus afin d'améliorer encore les gains.

Au niveau **système d'information**, on cherchera à :

- **Améliorer la qualité des données**, et en particulier des données de référence (le coût de non-qualité des données est estimé à plus de 600 milliards de dollars par an aux États-Unis)[1], notamment par rapport aux applications décisionnelles : éviter les données doublonnées, les données erronées ou incomplètes, les données non conformes…
- **Diminuer les coûts de gestion** liés aux données de référence. Il s'agit par exemple de limiter le nombre d'opérations de gestion comme les saisies, les corrections, les sauvegardes…
- **Assurer la cohérence** des différentes briques du **système d'information** : cohérence des données d'une application à l'autre, données à jour dans les applications…
- Rendre les **données plus accessibles et disponibles** : un accès bien identifié et peu de délai entre modification et mise à disposition.
- Assurer la **réactivité du système d'information** par rapport aux changements (du marché, des règlements, des organisations) : modifications dans une référence propagées aux applications du système d'information.
- Répondre aux **besoins de non-discrimination** vis-à-vis des contraintes réglementaires : accès, stockage, historisation, gestion de version…

Besoins

Les tableaux 1.1 et 1.2 listent respectivement les principaux besoins métier et système d'information associés aux données de référence, en les illustrant de quelques exemples et d'un indice de pertinence de l'utilité du MDM sur une échelle de 1 à 3.

1. Source : The Data Warehouse Institute (TDWI), in « *Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data* », 2002.

Tableau 1.1 — Besoins métier

Besoins	Exemples	Pertinence et support du MDM pour répondre au besoin
Disposer d'une **vue à 360°**, vue unifiée et cohérente des données de référence (en lecture, mais aussi en saisie)	– Disposer d'une vue à 360° du client. – Disposer du catalogue de l'ensemble des offres et prix. – Faciliter les recherches de données. – Homogénéiser et tenir à jour les données de référence.	***
Améliorer les processus opérationnels et décisionnels en disposant de données de référence de meilleure qualité pour les applications décisionnelles ou opérationnelles	– Disposer d'adresses fiables afin de minimiser les NPAI. – Optimiser la trésorerie (rejets de prélèvements bancaires). – Appliquer la bonne tarification : prix des produits et services, taxes.	**
Anticiper l'évolution des données pour prise en compte différée par le SI	Anticiper les changements d'organisation.	**
Disposer de données de référence à jour et non obsolètes au moment de leur utilisation	– Disposer de données externes à jour. – Disposer des données dans le bon état (données obsolètes inactives).	**

Tableau 1.2 — Besoins SI

Besoins	Exemples	Pertinence et support du MDM pour répondre au besoin
Simplifier les échanges (de données de référence)	– Automatiser les échanges entre entités organisationnelles. – Mutualiser les échanges afin d'éviter de redévelopper les interfaces à chaque besoin (fichiers de la poste, données météo…).	**
Assurer la cohérence des données de référence entre applications pour limiter ou faciliter les transcodages (de code ou d'identifiant)	Harmoniser les modèles entre progiciels. Résoudre les incohérences.	***
Assurer la qualité des données pour les applications consommatrices (notamment décisionnelles)	– Disposer des données de qualité pour applications.	**
Réduire les délais et les coûts de traitement liés aux modifications des données de référence, répercuter les modifications de données à l'ensemble des applications utilisant ces données	– Répercuter simplement et rapidement les modifications des organisations. – Diffuser la mise à jour du catalogue des offres à tous les SI supports des canaux de communication et canaux de vente.	***
Mutualiser l'acquisition des données de référence externes	Adresses, communes, banques, données météo…	***
Tracer les modifications et gérer les versions des données de référence	Satisfaire aux contraintes réglementaires (ART, CRE…)	***
Disposer d'une infrastructure pour gérer les données de référence	Démarche analogue EAI. Exemple : implémentation d'une infrastructure EAI, ESB, etc.	***
Accéder directement à des données éparpillées sur plusieurs bases ou tables	Besoin spécifique d'accès aux données.	***
Séparer les fonctions de gestion des données des fonctions métier	Fonctionnalités de gestion se retrouvant dans des applications opérationnelles.	***

1.1.7 Résumé des enjeux et besoins, importance de la GDR

Le tableau 1.3 synthétise ces enjeux et besoins.

Tableau 1.3 — Résumé des enjeux et besoins associés aux données de référence.

Besoins/enjeux	Métier	Système d'information
Cohérence globale	Aligner le vocabulaire et les définitions pour mieux partager en interne et avec les partenaires de l'entreprise.	Normaliser les définitions, les codes et les formats permettant un échange et une intégration mieux définis de SI internes ou étendus.
Unicité	Garantir l'identification d'un client, d'un produit, d'un fournisseur ou d'autres objets, afin d'offrir le meilleur niveau de service ou d'avoir la meilleure connaissance possible.	Garantir l'identification d'un objet métier afin de garantir la transcodification de cet objet avec l'ensemble des applications (transactionnelles ou décisionnelles) ainsi que minimiser la duplication d'information.
Visibilité/Disponibilité	– **Mieux utiliser les capacités du système d'information** grâce à des données de référence bien définies et partagées. – **Disposer de données de référence fiables et à jour** lorsqu'elles sont utilisées.	Offrir la bonne donnée au bon moment pour tous les processus outillés par les applications d'entreprise. Éviter les effets « bouchon » des chaînes batch « d'antan ».
Productivité/Agilité	– **Améliorer les processus** grâce à des données de référence de qualité et non redondantes. – **Réduire le temps des cycles métier** liés aux données de référence. – S'assurer face aux changements de marché ou de réglement.	– **Diminuer les coûts** de gestion liés aux données de référence : une seule saisie/validation. – **Assurer une meilleure agilité** du système d'information par la définition d'une couche *middleware* intégrant les données de référence.
Contraintes réglementaires	Mettre en œuvre les recommandations internes, respecter les contraintes légales, nationales ou supranationales (CNIL, Bâle II, Sox, Art. 29…).	**Répondre aux besoins de non-discrimination** : accès, stockage, historisation, gestion de version…

Besoins/enjeux	Métier	Système d'information
Qualité	Offrir aux processus métier des données propres et permettant le déroulement dans les meilleures conditions des activités d'entreprise.	**Améliorer la qualité des données de référence** : éviter les données erronées ou incomplètes, les données non conformes…
Sécurité	Garantir l'entreprise vis-à-vis de perte financière ou des risques d'image.	– Assurer les bons niveaux de cloisonnement au niveau de la largeur (l'ensemble des objets ou *data set*) comme au niveau de la profondeur (précision au niveau des attributs ou *data model*). – Sécuriser les nouveaux projets, identifier les sources des données de référence.

La cohérence des données de référence est indispensable en particulier pour les processus de gestion transverses. Malheureusement, dans des infrastructures systèmes vastes et hétérogènes, la discordance des données de référence se traduit par la redondance des données, voire par des erreurs de processus ou encore d'analyse conduisant à des surcoûts ou à de mauvaises décisions en matière de gestion.

Or, homogénéiser les données de référence entre tous les systèmes d'un environnement informatique distribué a toujours été un exercice difficile. Cela suffit à expliquer que les succursales d'une entreprise travaillent souvent indépendamment de la maison mère, se mettant alors à l'écart d'opportunités de synergie ou d'économie d'échelle. Les fusions et les acquisitions posent aussi de gros problèmes en ce qui concerne la cohérence des données de référence, de même que les *splits* d'entreprise (dérégulation ou antitrust par exemple).

Si, sur le plan technique, les entreprises parviennent à intégrer les nouvelles solutions logicielles provenant d'une acquisition, elles intègrent en revanche difficilement les processus de gestion à cause d'incompatibilités fondamentales entre les processus eux-mêmes et entre les modèles de données.

En définitive, la gestion harmonieuse et maîtrisée des données de référence permet la mise en place, dans toute l'entreprise, de processus de gestion et de systèmes d'analyse fiables et homogènes, offrant ainsi à toutes les personnes impliquées la possibilité d'avoir accès aux mêmes informations et connaissances. C'est pourquoi une solution permettant à la fois la consolidation des données de base et l'accès à des données globalement cohérentes sur tous les systèmes confère un avantage concurrentiel décisif.

La gestion des données de référence (GDR) doit donc être placée au cœur de chaque stratégie d'entreprise quel que soit son secteur d'activité, du commerce au secteur financier, en passant par la production, la distribution, le service. Par souci

de compétitivité, il est nécessaire d'investir du temps, des ressources humaines et financières dans une stratégie de gestion des données de référence afin de garantir des informations de qualité.

1.2 EXEMPLES D'ENTREPRISES

Les exemples suivants sont issus de l'expérience des auteurs. Ces exemples seront le fil rouge de cet ouvrage ; les solutions mises en place seront analysées en deuxième partie et les méthodes et organisations dans la troisième partie.

1.2.1 Un grand distributeur

La société X, en tant qu'entreprise de grande distribution, est particulièrement sensible à la connaissance et à la maîtrise de la famille de ses référentiels dits « marchands » (produits en premier lieu, mais aussi fournisseurs, contrats ou assortiments, voir figure 1.9).

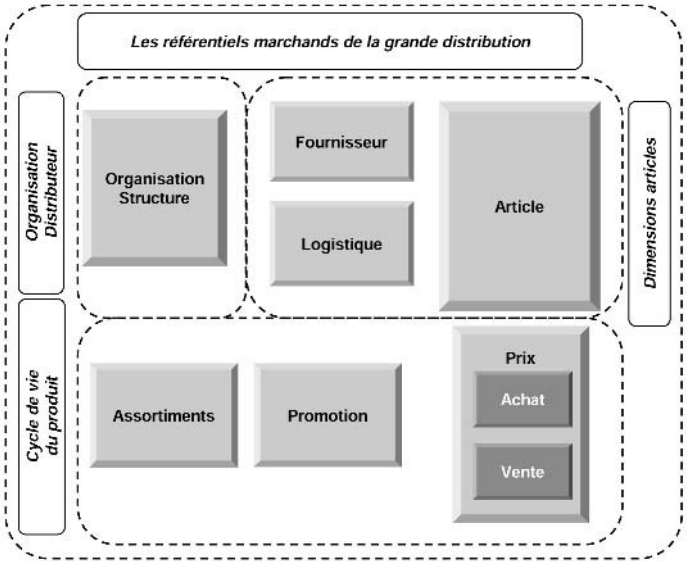

Figure 1.9 — La famille des référentiels « marchands »

Bien que sensibles, ces données n'ont pas été initialement gérées en englobant l'ensemble des contraintes. En effet, la logique d'évolution des SI de la grande distribution a souvent été dictée par la recherche de bénéfices rapides, générant ainsi une juxtaposition de couches applicatives et des lourdeurs d'ensemble des processus (resaisies multiples, systèmes incohérents, pas de maîtrise des erreurs).

Confrontée à de lourds impacts sur les solutions consommatrices et les processus métier dépendants, l'entreprise a cherché un moyen de maîtriser la gestion, la qualité et la diffusion de ces données.

Dans le même temps et suite au rachat par la société X de la société A, les SI et les équipes des deux entreprises ont fusionné.

Le groupe s'est ainsi naturellement saisi du sujet « référentiel marchand » car cette problématique générale entraîne de lourds frais de gestion et il semblait logique de rationaliser et de maîtriser l'ensemble du processus d'acquisition d'une donnée directement liée à la maîtrise et à la connaissance des achats, première source de revenus (de ce type d'entreprise) par la maîtrise des marges arrières (facturation, décisionnel, performance fournisseurs en sont donc la clé).

La logique de la société X n'est pas de déployer une stratégie globale pour l'ensemble de ses référentiels. Le groupe répond avant tout à des besoins métier. La construction des référentiels marchands bénéficie donc de son propre socle et de ses propres instances (comité de gouvernance, centre de compétence). Plusieurs projets ont donc été initiés pour maîtriser l'ensemble de la chaîne référentielle (préréférentiel, accès B2B, modélisation fiche produit, *workflow* d'enrichissement, *workflow* de contrôle, indicateurs de suivi, gestion des erreurs, référentiels de consommation…).

La société X s'est appuyée sur un premier déploiement pays réussi afin de généraliser sa démarche de référentiel. Cette réussite a d'ailleurs précédé la plupart des projets ainsi que la constitution du centre de compétence référentiel et a permis d'initier la dynamique dans chaque pays.

1.2.2 Un producteur

La société Y a lancé un plan de transformation de son système d'information. Ce plan visait la rationalisation du système d'information par la mise en œuvre d'applications majeures en soutien des grands processus métier de l'entreprise. Ces applications majeures ont été conçues sous la forme de « CoreModel » et déployées localement sous celle de « CoreSystem ».

Ces applications ont nécessité la mise en place d'une organisation et d'une méthode de déploiement spécifiques (centre de réalisation unique et déploiement local des solutions par plaque continentale). Les référentiels répondent à la même logique de conception, réalisation et déploiement.

La société Y a identifié treize référentiels principaux dans son plan de transformation du système d'information. Le référentiel client, de par sa nature stratégique (réorientation du système d'information vers les clients plutôt que vers les produits) a été identifié comme le premier référentiel à mettre en œuvre. Un référentiel de paramètres (tables et listes de valeurs) suivra, afin de participer à la constitution du socle mutualisé des référentiels.

La société Y s'est dotée d'une organisation responsable de la définition des règles et du suivi des projets suivant l'axe référentiel. Cette organisation s'étend du niveau

stratégique (*Governance Committee*) au niveau opérationnel (*Stewardship Comittee*). Les métiers sont très fortement impliqués.

1.2.3 Un fournisseur

La société Z, au travers de son entité « Innovation », a très rapidement été sensibilisée à la problématique de la gestion des référentiels. En effet, elle est en charge, pour le compte du groupe, de la gestion des projets d'*eBusiness* et d'*eProcurement* sur les processus de commande, de livraison et de facturation. Ces processus s'appuient sur les données clés que sont les fournisseurs, leurs catalogues, produits et services, les données d'identification (code DUNS)… L'objectif de la Direction Achat Groupe est de bénéficier d'un outil leur permettant de rationaliser les achats au travers d'un référentiel commun au groupe sur les données fournisseurs.

1.3 POURQUOI METTRE EN ŒUVRE UNE GESTION DES DONNÉES DE RÉFÉRENCE ?

D'une manière générale, à défaut d'une gestion adéquate des données de référence, que constate-t-on dans les entreprises ?

- La plupart des applications gèrent en local leurs données de référence avec des modèles parfois différents et des données incohérentes. On note des **saisies multiples et la génération potentielle de doublons entre applications.** Exemple : les organisations sont définies au sein d'un progiciel RH, mais ressaisies dans chaque Direction d'une entreprise.
- Il est difficile de récupérer certaines données de référence externes (par exemple, les adresses, les codes bancaires).
- Certaines interfaces ou modules de chargement de données sont développés plusieurs fois (par exemple, le chargement des fichiers de la poste).
- Les protocoles d'échanges de ces données de référence sont hétérogènes (courrier électronique, échanges de fichiers…).
- Les mises à jour de ces données de référence ne sont pas coordonnées, souvent en décalage.
- Il existe des écarts pour une même donnée de référence (valeur ou liste de valeurs) utilisée par plusieurs applications. Par exemple, un même fournisseur codifié de manière différente dans deux systèmes donne lieu à une analyse de deux chiffres d'affaire entravant une bonne négociation.
- Les applications s'échangent des données de référence au sein d'objets métier et doivent souvent mettre en place des transcodifications entre leurs valeurs de référence et celles utilisées par d'autres.
- **Lors de la mise en œuvre d'un nouveau projet, il est souvent difficile d'identifier les sources des données de référence.** Exemple : plusieurs appli-

cations gèrent des contrats associés à des clients. Quelle est la référence du contrat du client Z ?

- Chaque nouveau projet permet d'améliorer la donnée au moment du chargement initial mais ne poursuit pas par la suite.

Ce constat fait clairement apparaître le besoin d'une « gestion des données de référence » qui assure les fonctions suivantes :

1. Construire une vision unique et partagée de la donnée

Une vision unique de la donnée de référence doit être définie pour l'ensemble de l'entreprise. Elle peut être détaillée et spécialisée suivant le contexte, mais ces détails ne doivent pas faire partie de la donnée de référence.
La donnée de référence ainsi définie doit être partagée par l'ensemble des processus qui l'utilisent.

2. Intégrer la donnée en provenance de systèmes sources

La donnée de référence doit pouvoir provenir de tout type de système source. Cela implique qu'avant d'être intégrée, elle doit être transformée pour correspondre à la vision unique partagée par l'entreprise. Cela implique aussi une vision claire et documentée des droits et de la propriété sur la donnée à l'échelle du système d'information.

3. Intégrer la donnée de référence dans les processus de l'entreprise

La donnée doit être diffusée et mise à disposition de manière à ce que l'ensemble des processus de l'entreprise puissent y accéder, que ce soient les processus opérationnels et décisionnels.

4. Améliorer la qualité de la donnée de manière continue

La qualité de la donnée doit être maintenue aussi bien lors de sa création que lors de la vie de la donnée de référence. Cette qualité correspond aux concepts d'unicité, d'exactitude, de complétude et de cohérence avec l'ensemble des données SI.

5. Créer, diffuser et maintenir les métadonnées de données de référence

Les données de référence doivent être décrites de manière complète et pertinente. Cette description doit être maintenue et surtout accessible par l'ensemble de l'entreprise.

Cette gestion doit s'appuyer en outre sur les trois principes suivants (cf. figure 1.10) :

- Identification de la source qui fait foi pour un certain nombre de données de référence transverses.
- Mutualisation de l'administration de ces données de référence, en mettant en place les outils adaptés à leur gestion.
- Mise à disposition ou diffusion de manière coordonnée de ces données de référence aux différents consommateurs.

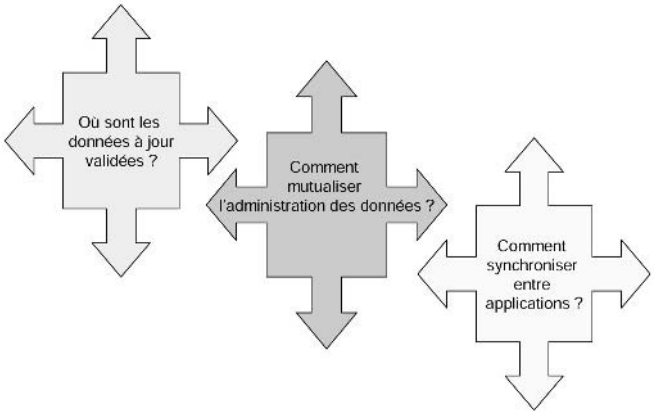

Figure 1.10 — Les principaux problèmes liés aux données de référence

Les bénéfices que l'entreprise doit en attendre sont multiples :

- amélioration du fonctionnement des processus opérationnels et décisionnels grâce :
  - à la mise en place d'une organisation garantissant la qualité des données diffusées (par exemple, changement d'un libellé de voie, prise en compte d'un nouveau code postal…) ;
  - au partage des mêmes données de référence par les différentes applications et à une meilleure coordination de leur diffusion, afin de réduire notamment les rejets lors de l'import des données provenant d'une autre application ;
  - à une augmentation de la fréquence de diffusion des données de référence.
- diminution des coûts d'interface pour les nouvelles applications (ou pour une application existante ayant besoin d'un nouvel accès) grâce à la mutualisation de certaines interfaces (accès à des sources externes de données, comme les données météo, les codes communes INSEE…) et à l'utilisation de formats de fichier plus adaptés ;
- diminution des coûts d'administration des données de référence, grâce à :
  - la centralisation et la mutualisation de leur récupération et d'une partie de leur administration (chaque application devant continuer à prendre en charge sa propre mise à jour par rapport au référentiel) ;
  - la possibilité d'automatiser le chargement dans les applications clientes grâce à une meilleure structuration des données au niveau du référentiel (modèle de données) ;
  - la réduction des transcodifications, grâce à l'utilisation d'une même source de données.
- limitation des impacts sur les applications en cas d'évolution du format ou du mode de transmission des données de référence, grâce à la centralisation de la récupération de ces données ;

- plus grande réactivité du système d'information en cas d'évolution d'une valeur ou d'une liste de référence, en particulier grâce au partage des mêmes valeurs. Cela évite les transcodifications, *via* un paramétrage des applications ou des pivots. De plus, la préexistence d'une source identifiée prenant en charge la gestion et la diffusion de ces données facilite l'intégration d'une nouvelle application (ou un nouveau besoin d'accès par une application existante).

En définitive, on pourra sommairement simplifier notre propos et résumer en affirmant que les gains attendus reposent sur trois axes principaux :

- **L'amélioration des processus**, au premier rang desquels les processus référentiels, suivi des processus métier dont participe la donnée.
- **L'amélioration des capacités analytiques** de l'entreprise (décisionnelle), par la prise en charge de la qualité des données et les capacités de corrélation des objets inhérentes au MDM.
- **L'amélioration de la communication**, que ce soit les échanges entre les applications de l'entreprise et donc les processus et métiers de celles-ci, mais également les capacités de partage d'information avec les partenaires.

Chacun de ces axes n'est pas exclusif et vous pourrez les additionner au sein de vos initiatives MDM. Nous retrouverons, dans la deuxième partie, des exemples de projets répondant à chacun de ces axes.

En résumé

Une donnée de référence est une **donnée du système d'information, partagée et utilisée sans modification par les applications**. Ces données sont les éléments de base de la gestion de toute entreprise.

On distingue les données « **maître** », qui sont des objets « cœur de métier » d'un domaine fonctionnel, les données « **constitutives** » qui caractérisent en général des données maître et les données « **paramètre** » : tables de valeurs ou nomenclatures. Leur gestion relève du domaine de la gestion des données de référence (GDR) ou MDM (*Master Data Management*). La cohérence, l'unicité, la fiabilité, la disponibilité, la sécurité sont les principaux besoins auxquels les données de référence doivent satisfaire pour être mieux partagées, diminuer les coûts, améliorer les processus et optimiser les ressources du système d'information.

2

La donnée et ses dimensions de valeur

Objectif

Il s'agit ici de montrer :

– en quoi les données contribuent aussi à la création de valeur à travers les bénéfices attendus d'une véritable gestion de leur qualité, domaine de gestion spécifique et étroitement lié à la gestion des données ;

– comment améliorer la qualité des données et identifier les objectifs à atteindre pour chacun des critères qualité en fonction de leur utilisation ;

– en quoi les métadonnées participent à cette démarche qualité.

2.1 DONNÉES ET VALEUR

La donnée est une ressource, un actif de l'entreprise, au même titre qu'un bien matériel, et à ce titre :

- elle peut être valorisée ;
- elle a un cycle de vie.

On peut définir la **valeur d'une donnée** comme la différence entre les bénéfices obtenus et les coûts engendrés (de stockage, d'administration, de maintenance…). Pour une valeur optimale pour l'entreprise, la donnée devra être de qualité, car toute source de non-qualité (que nous détaillerons dans la suite) engendre à la fois des bénéfices moindres et des coûts supplémentaires.

On remarquera cependant qu'il peut être mal aisé d'identifier les bénéfices d'une bonne qualité des données. En effet, les métiers sont obnubilés par les processus et cherchent à juste titre à les améliorer pour générer de la valeur. Mais les évaluations de gains sont uniquement centrées sur ces processus métier et ce qui supporte ou structure ces processus, à savoir les données, est peu pris en compte.

Une analyse des risques de non-qualité peut permettre de démontrer la valeur de la donnée.

Outre l'amélioration de la qualité, une gestion maîtrisée des métadonnées est indispensable pour traiter la donnée comme un actif valorisé.

2.2 QUALITÉ DES DONNÉES

2.2.1 Principaux concepts

Les données jouent un rôle croissant dans les processus opérationnels et dans les choix stratégiques des entreprises. Les entreprises, qui ont souvent fortement amélioré leurs processus, doivent maintenant agir directement sur les données sous-jacentes qui les structurent pour espérer de nouveaux gains.

De même pour le pilotage, la qualité des résultats fournis par les systèmes décisionnels dépend directement de la qualité des données qui leur sont fournies. Les traitements en « redressement qualitatif » au niveau des systèmes décisionnels sont peu efficaces. Une amélioration n'est réellement envisageable qu'en ayant une meilleure maîtrise de la chaîne de l'information, c'est-à-dire en se rapprochant du point où l'information est créée ou garantie.

La mauvaise qualité des données peut donc engendrer de multiples conséquences néfastes pour le bon fonctionnement de l'entreprise, son économie, son image et son évolution.

Des données de qualité médiocre perturbent l'activité d'une entreprise au jour le jour, occasionnant non seulement des surcoûts de fonctionnement, mais aussi l'insatisfaction des clients et donc des pertes à court terme.

Ces perturbations affectent la réputation de l'entreprise et diminuent son potentiel de croissance. Enfin, la prise de mauvaises décisions stratégiques consécutives à une qualité de données déficiente peut engager l'entreprise sur de mauvaises voies. Par exemple, la connaissance des clients et des produits étant désormais devenue un enjeu essentiel, le besoin de contrôle et d'amélioration de la qualité des données clients et produits qui en découle contribue depuis peu à faire émerger des outils et des offres génériques, ou déclinées par métiers.

La norme ISO 8402[1] définit ainsi la qualité : « l'ensemble des caractéristiques d'une entité qui lui confèrent l'aptitude à satisfaire des besoins exprimés ou implicites ». Plus simplement, on parlera du : « **degré d'adéquation à l'usage que l'on en fait** ».

Ainsi, dans le contexte de la gestion des données de référence : la qualité des données est l'aptitude de l'ensemble des caractéristiques intrinsèques des données (unicité, exhaustivité, fraîcheur, disponibilité, cohérence fonctionnelle, cohérence technique) à satisfaire des exigences internes (pilotage, prise de décision…) et des exigences externes (réglementations par exemple) à l'organisation.

2.2.2 Les critères de la qualité des données

Relativité de la qualité des données

Pour faire une offre promotionnelle à un client à l'occasion de son anniversaire, il est indispensable de connaître sa date de naissance alors que pour mener une étude marketing sa tranche d'âge suffira.

Imaginons en effet un référentiel client contenant les âges de *n* clients. Supposons que les âges soient connus à un an près. Cet ensemble référentiel client ne pourra pas servir à souhaiter à chaque client son anniversaire à la bonne date. Pour cet usage, on peut considérer que les données sont de très mauvaise qualité. Par contre, les données sont suffisantes pour une étude marketing.

Cet exemple illustre pourquoi des données jugées de mauvaise qualité par les personnels en charge de la relation client peuvent être jugées de bonne qualité pour effectuer des études marketing.

On notera également l'intérêt de l'établissement de métadonnées par le système d'information décrivant la précision de celles-ci, afin d'aider les utilisateurs à pouvoir donner eux-mêmes la précision du résultat de leurs études.

1. http://www.iso.org/iso/fr/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=20115. On notera que la norme ISO 8402:1994 est maintenant remplacée par la norme ISO 9000:2005.

Les critères intrinsèques

L'unicité

L'unicité est le fait qu'une entité du monde réel ne soit représentée que par un seul et unique objet métier au sein de l'entreprise. Cet objet ne répond donc qu'à un unique identifiant.

Cette qualité garantit que des objets de l'entreprise ne pourront pas être confondus. Il s'agit donc d'éviter les doublons, soit à l'intérieur d'une même application (un répertoire MDM par exemple), soit entre différentes applications.

Il existe deux façons d'établir l'unicité d'une donnée. La première, dite « déterministe », repose sur des règles de rapprochement simple sur des champs discriminants précis (par exemple, le même identifiant ou le même numéro de téléphone ou le même triplet « nom/prénom/adresse »).

La seconde est dite « probabiliste » et repose sur un processus plus fin et des algorithmes de rapprochement prenant en compte non seulement ce que les objets comparés portent comme information mais aussi les fréquences ou approximations compatibles avec l'ensemble des informations détenues au sein de l'ensemble des objets comparés.

Exemple : l'objet ID-4937 « Jean Dupont au 32 rue de la venaison » et l'objet B ID-2845 « Jean Dupond au 32 rue de la venaison » sont, avec une grande probabilité, des doublons.

La complétude

La **complétude** est la présence de valeurs de données significatives pour un ou des attributs, un ou des objets.

On doit en principe préciser sur quel(s) élément(s) porte(nt) la définition de la complétude. En effet, on peut suivre la complétude au niveau d'un objet (tel attribut est renseigné ou non) ou au niveau du jeu de données.

Exemple au niveau objet : dans le cadre des fournisseurs, on s'intéresse particulièrement à la complétude du SIRET pour les différentes catégories d'entreprises.

Exemple au niveau du jeu de données : les 80 000 clients du segment UHNWI (*Ultra-High-Net-Worth Individuals*)[1] font tous partie du référentiel Prospects de Roll Royce.

L'exactitude

Une donnée est « **exacte** » si la valeur des attributs de l'entité concernée est égale à la grandeur qu'elle est censée représenter (dans le monde réel). Cette notion englobe donc deux aspects : la **précision** et la **validité**.

En effet, une information peut être déclarée valide car elle répond aux contraintes fonctionnelles de validation sans pour autant être précise.

---

1. « Rich spurn ultra-luxury cars », *The Sunday Times*, 5 novembre 2006, http://business.timesonline.co.uk/tol/business/article1086869.ece.

Pour être exacte, la granularité d'information doit juste répondre à la règle d'adéquation à l'usage déjà citée.

Une donnée « relativement précise » (valeur approchée) ne sera pas exacte au sens strict, mais pourra néanmoins être utilisée dans certains cas si l'imprécision est maîtrisée et correspond au niveau de granularité requis (consommation annuelle moyenne approchée, par exemple).

Exemple : le code postal d'une ville fait partie de la liste des codes postaux de la poste. Celui de Boulogne Billancourt est 92100 et correspond bien à la ville portée dans le champ adresse de l'objet.

Autre exemple : Le capital investit des UHNWI est supérieur ou égal à 30 M$. Un référentiel « prospect » ne dit pas cependant si John Hoover Senior est à la tête de 30, 50 ou 100 millions de dollars.

La conformité

La **conformité** d'un ensemble de données est le respect par celles-ci d'un ensemble de contraintes. Par exemple, l'identifiant d'un équipement doit commencer par deux lettres suivies de trois chiffres.

En fait, la conformité peut être vue comme une sous-dimension de l'exactitude : des données intégralement exactes sont conformes. Néanmoins, cette dimension s'avère d'une importance pratique capitale car c'est elle qui offre le plus d'opportunités de contrôles sur les données (contrôles de conformité). Les contrôles de conformité constituent autant d'étapes permettant d'éliminer peu à peu des données inexactes (ou suspectées de l'être). La conformité par rapport à un ensemble de contraintes exprimées ne constitue toutefois pas en général une garantie absolue d'exactitude.

La cohérence

Cette notion de **cohérence** dépend à la fois de critères intrinsèques et des services.

Au niveau intrinsèque, elle est relative à l'absence d'informations conflictuelles au sein d'un même objet (par exemple, une incohérence serait détectée si un « prix actuel » d'un produit est supérieur au « prix maximum » de ce même produit). Mais cette notion existe aussi au niveau service : les valeurs d'une instance d'un objet métier ne sont pas en conflit avec les valeurs d'une autre instance ou d'une instance d'un autre objet.

Exemple : Éric Dujardin est le fils d'Hélène Dujardin. L'objet client « Éric Dujardin » ne peut pas porter la valeur DateNaissance=13/04/1991 quand l'objet client « Hélène Dujardin » porte la valeur DateNaissance=25/04/2005.

À l'échelle du système d'information on remarquera que des instances d'objets métier peuvent être incohérentes le temps nécessaire à la resynchronisation d'ensemble du système d'information. Des processus et une gestion d'états complexe peuvent potentiellement être mis en œuvre pour gérer ces statuts de propagation.

L'intégrité

L'intégrité concerne les relations entre objets. Les relations importantes entre objets sont-elles toutes présentes ?

Exemple : toute facture doit être associée à une commande. Si une facture n'a pas de référence vers une commande, c'est un problème d'intégrité.

La figure 2.1 schématise ces principaux critères de qualité.

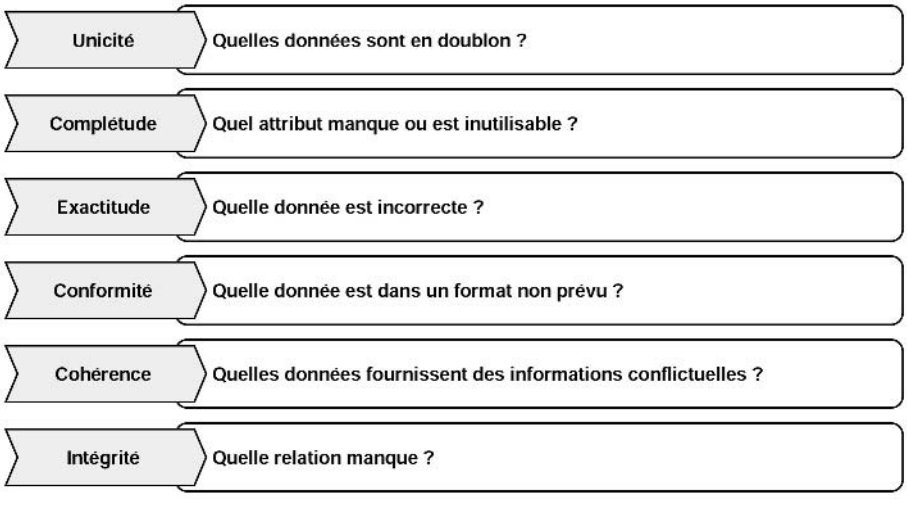

Figure 2.1 — Principaux critères de qualité

Les critères de services

L'actualité

De nombreuses dimensions qualité concernent le rapport entre les données et le temps (**actualité**) :

- L'obsolescence est le fait que la valeur de la donnée, autrefois exacte, ne l'est plus suite à un changement (dans le monde réel) de l'objet représenté.
- L'obsolescence peut aussi porter sur la représentation d'une donnée qui a été modifiée.
- Une valeur de donnée est à jour si elle est correcte en dépit d'un écart possible avec la valeur exacte, due à des changements liés au temps ; une donnée est périmée à la date *t* si elle est incorrecte à cette date mais était correcte aux instants précédant *t*. L'actualisation est le degré mesurant à quel point une donnée en question est à jour (par exemple, l'âge ne devient obsolète qu'à la date anniversaire).

Exemple : plus de 3 millions de foyers déménagent chaque année en France[1], ainsi on considère qu'environ 10 % des adresses d'un fichier client deviennent obsolètes chaque année.

1. Source INSEE

Les utilisateurs présentent une grande sensibilité aux données mises à jour trop tardivement, situation qui leur impose l'utilisation d'historiques incomplets ou le recours à des données trop anciennes. Ce constat met en évidence le besoin de :

- créer et mettre à jour les données suffisamment souvent (fine granularité temporelle) ;
- mettre à disposition des utilisateurs le plus vite possible.

Ainsi, on préférera donc le mode message en temps réel ou « fil de l'eau » pour répondre aux contraintes de temps dans la sphère transactionnelle comme dans la sphère décisionnelle.

Remarquons que ce critère est proche de la cohérence (entre applications) exposé dans le paragraphe précédent.

L'accessibilité

L'accessibilité est la dimension qualité qui concerne la facilité d'accès aux données.

Dans l'univers MDM, cela signifie que les services de données sont calibrés en fonction de leur utilisation et qu'ils existent souvent aussi bien en mode événement (déclenché à chaque mise à jour), qu'en mode requête (à la demande d'un processus consommateur) ou en mode *batch* pour des synchronisations en masse (pour le décisionnel par exemple).

La pertinence

La **pertinence** est la dimension qualité qui définit l'utilité d'une donnée.

Une donnée peut être accessible mais tellement détaillée que de nombreux attributs de l'objet proposé sont inutiles aux processus consommateurs. Une donnée doit être adéquate à son usage. Les services de donnée seront d'autant mieux utilisés que la granularité d'information dispensée correspondra aux besoins.

La compréhensibilité

La **compréhensibilité** est la dimension qualité associée à la question : « cette donnée est-elle compréhensible ? ».

Une donnée est compréhensible si chaque utilisateur, chaque processus, chaque application trouve facilement la bonne information parmi les attributs disponibles d'un objet. C'est le cas si celui-ci est clair et que l'alignement sémantique de l'ensemble des concepts entre tous les dépositaires (humains ou informatiques) a été réalisé et documenté.

Le tableau 2.1 reprend ces critères et leur définition.

Tableau 2.1 — Principales dimensions de la qualité des données

Niveau	Dimensions qualité des données	Définitions ou questions associées
Intrinsèque	Unicité	Des données uniques n'ont pas de doublon.
	Complétude	Un ensemble de données est complet quand toutes les valeurs prévues sont renseignées.
	Exactitude	Des données sont exactes si leurs valeurs le sont.
	Conformité	Des données conformes respectent un ensemble de contraintes.
	Intégrité	Des données intègres sont correctement liées.
Service	Cohérence	Des données sont cohérentes quand il n'y a pas de conflit.
	Accessibilité	L'accès aux données est-il aisé ?
	Actualité	Les rapports entre les données et le temps sont-ils optimaux ?
	Pertinence	Les données sont-elles utiles ?

Les critères de sécurité

Un peu à la marge de la qualité, rappelons que la sécurité est l'une des premières règles de bonne gestion des données avec les principaux critères suivants :

- Disponibilité : aptitude du système à remplir une fonction dans des niveaux de service définis.
- Intégrité : légitimité de toute modification de donnée.
- Confidentialité : accès limité à qui de droit pour les besoins de service.
- Traçabilité : conservation des opérations effectuées et de leurs auteurs.

2.2.3 Quels objectifs de qualité des données ?

En fonction des utilisations de la donnée, on détermine les critères qualité primordiaux à contrôler. Ensuite, on détermine les attributs de la donnée permettant de mesurer les critères qualité. Enfin, on spécifie le niveau minimal de qualité requise pour chaque critère retenu.

Il est en effet impératif de rendre mesurable le niveau de qualité. Nous reviendrons sur les indicateurs opérationnels de suivi dans la troisième partie de ce livre.

Dans un contexte MDM, les objectifs sont plus nombreux et plus diversifiés que dans le simple univers décisionnel. La granularité d'information et de suivi qualité doit répondre à tous les objectifs des processus consommateurs.

Par exemple, pour mesurer la fraîcheur de l'information, on peut choisir de suivre les données « dernière date de mise à jour » et « date du jour » ou bien « date de création » et « date du jour ». En MDM on les suivra certainement des deux manières.

Autre exemple : est-il admissible de ne pas cibler un prospect dans une campagne marketing ou d'envoyer à un client un courriel relatif à une offre qu'il a déjà souscrite ? Dans le premier cas, la complétude des informations sur les prospects est très importante. Dans le second cas, il est obligatoire de pouvoir croiser le nom d'une personne entre un fichier de prospection marketing et un fichier des offres souscrites. Dans un contexte MDM, on portera le niveau d'information au degré nécessaire de telle façon que les deux cas soient réalisables.

2.2.4 Amélioration de la qualité des données

Approche globale

L'amélioration de la qualité des données est une démarche continue. Elle commence dès l'analyse des sources de données, et se poursuit avec la préparation du chargement du référentiel, et consiste enfin en un suivi régulier de l'activité.

La figure 2.2 schématise les étapes à suivre pour améliorer la qualité des données :

- Valider le niveau de qualité sur l'existant (étude).
- Définir le niveau de qualité cible (étude).
- Atteindre le niveau de qualité cible (projet).
- Rester à ce niveau (production).
- Surveiller la qualité (production).

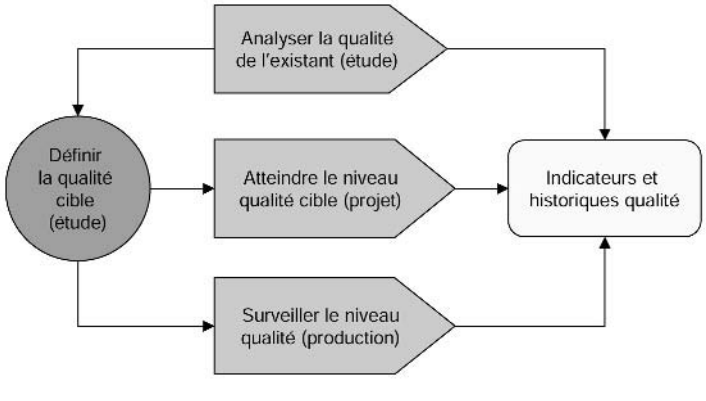

Figure 2.2 — Démarche globale des opérations d'amélioration de la qualité des données

Approche « nettoyage », ou data cleansing

Le « nettoyage » consiste à détecter et corriger des erreurs dans les données afin d'améliorer leur qualité. Il s'agit d'une activité en général réalisée en bout de chaîne, la correction des données lors de la saisie (à la manière du correcteur orthographique de Word) étant souvent négligée. Le « nettoyage » est typiquement réalisé dans le processus ETL, avant chargement des données dans un *datawarehouse*. Il peut aussi être mis en œuvre de manière curative sur des données existantes, avant migration vers une nouvelle application.

Le principal défaut de l'approche « nettoyage » est de ne pas assurer la prévention des erreurs futures. Toutefois, dans le cas de données relativement peu fréquemment créées ou modifiées, cette approche permet d'améliorer sensiblement et durablement la qualité d'une base de données. C'est également très utile pour migrer des données existantes de mauvaise qualité vers un nouveau système.

Approche « processus »

L'approche « processus » a pour objectif de prévenir l'introduction de données erronées dans un système d'information. On entend par « processus » toute la chaîne de traitements et d'opérations, de la création des données à leur destruction, en passant éventuellement par des modifications de leurs valeurs.

La figure 2.3 schématise ces deux approches de la qualité des données.

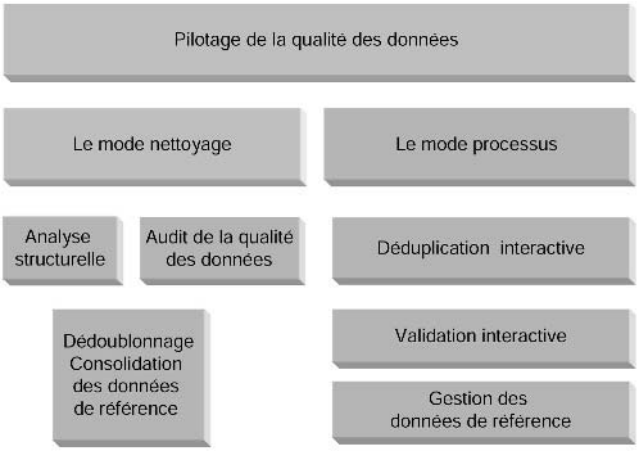

Figure 2.3 — Approches « nettoyage » et « processus » de la qualité des données.

Ces deux approches d'amélioration des données peuvent sembler opposées : l'approche « processus » se propose d'éliminer les causes de non-qualité à la source, tandis que l'approche « nettoyage » se propose de « dépolluer » a posteriori des données en défaut.

L'approche processus relève plutôt du champ d'action des systèmes informatiques d'acquisition et de mise à jour des données (généralement sous la responsabilité directe des utilisateurs), tandis que le « nettoyage » procède plutôt d'une culture statistique et *data mining* (moins immédiatement utilisables dans un cadre opérationnel). Remarquons toutefois que tous les contrôles à la saisie ont leurs limites et ils ne peuvent au mieux que représenter des données vraisemblables (pas forcément exactes ou complètes).

En réalité, ces deux approches présentent de nombreux points communs et sont complémentaires :

- Elles poursuivent le même objectif : améliorer la qualité.
- Elles nécessitent la mise en place d'outils de contrôle et de suivi de la qualité.
- Elles s'appuient fortement sur l'expertise métier.

2.3 PRINCIPALES CAUSES D'INCOHÉRENCE OU DE NON-QUALITÉ

Le tableau 2.2 qui suit indique les principales causes de non-cohérence ou de non-qualité.

Tableau 2.2 — Causes de non-cohérence ou de non-qualité

Causes d'incohérence ou de non-qualité	Description
Conflit sémantique	Aucun consensus n'a été trouvé sur la définition précise de la donnée d'entreprise pour tous les intervenants. Le même concept est interprété de manière différente.
Conflit de modèle	La donnée d'entreprise est figée dans sa sémantique mais aucun modèle ne permet d'en construire une représentation compréhensible et partageable par les différents intervenants qui l'utilisent. Le même concept est modélisé de manière différente (noms de champs différents ou construction différente).
Mode opératoire mal défini ou non respecté	La saisie ou la mise à jour de la donnée ne respecte pas une séquence adéquate ce qui introduit des états d'incohérence ou de non-qualité.
Applications introduisant des incohérences : référence non unique, données non à jour, etc.	Pas de référence (pas de point unique de vérité), saisies multiples non cohérentes, données non à jour à cause des échanges batch, etc.

2.4 EXEMPLES DE NON-QUALITÉ OU D'INCOHÉRENCE

2.4.1 Doublons

La figure 2.4 correspond à un exemple de génération de doublons.

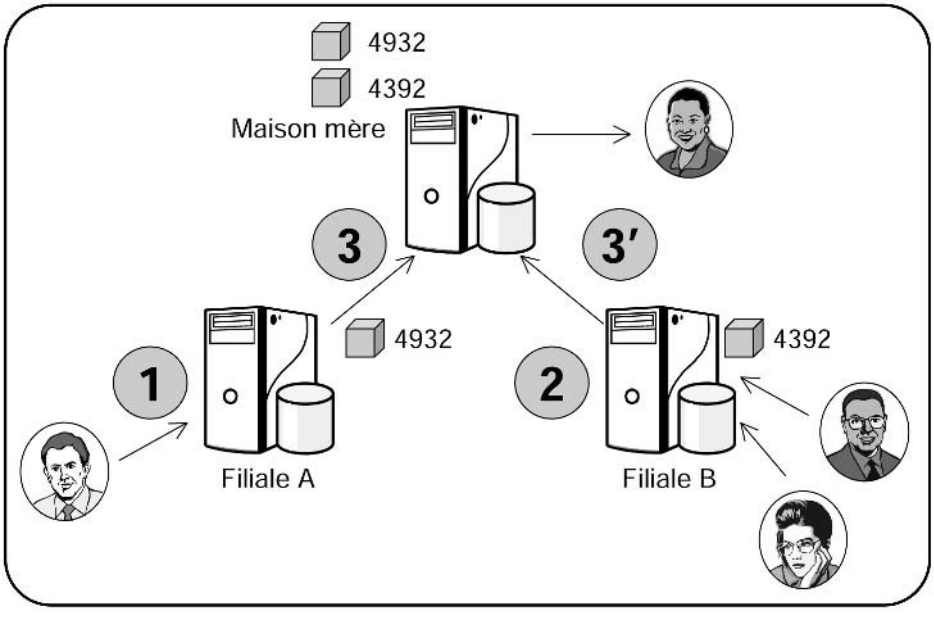

Figure 2.4 — Exemple de génération de doublons

- **Description**
  - (1) : Saisie par l'équipe filiale A du produit X sous la référence N° 4932.
  - (2) : Saisie par l'équipe filiale B du produit X sous la référence N° 4392. (vs 4932).
  - (3) et (3') : Transfert des données référentielles pour consolidation.
- **Problèmes** – Génération de doublons dans le système récepteur.
- **Causes** – La procédure de création de données maître est mal définie et/ou mal respectée.
  L'architecture n'a pas d'application maître (« point de vérité ») identifiée (cette notion est détaillée dans la suite de l'ouvrage).

2.4.2 Données incomplètes

La figure 2.5 donne un exemple de génération de données incomplètes.

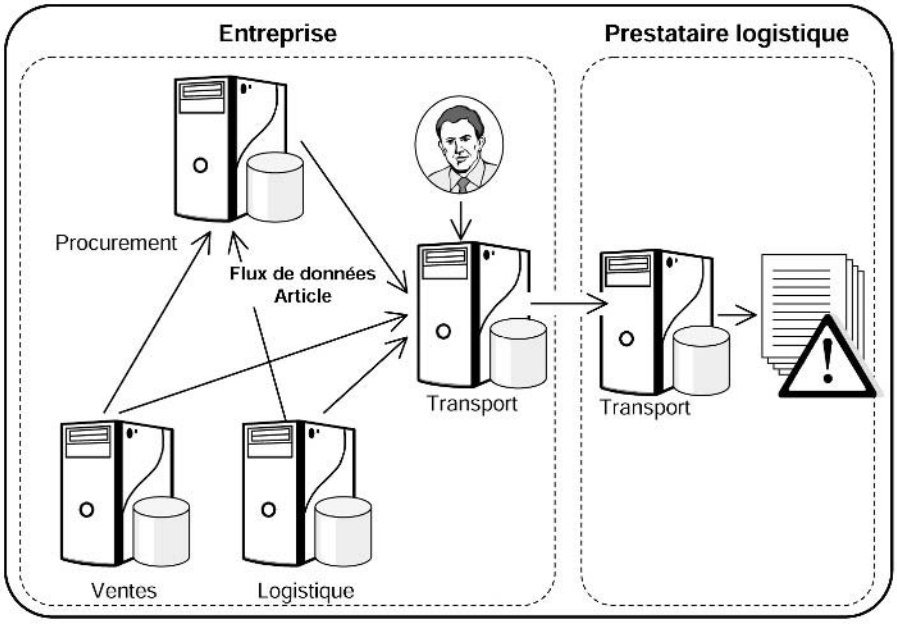

Figure 2.5 — Exemple de génération de données incomplètes

- **Description** – Une entreprise de logistique internationale opère des transports, pour le compte d'une entreprise cliente, entre l'entreprise et ses fournisseurs.
  Les données « Articles » descriptives des marchandises transportées sont de la responsabilité de l'entreprise cliente.
- **Problèmes** – La donnée est incomplète et ponctuellement des camions restent bloqués en douane pour cause de documents erronés.
- **Causes** – Les procédures de saisie de la donnée article sont réparties sur de multiples applications, générant une donnée parfois incomplète.

2.4.3 Données trop longues à générer

La figure 2.6 présente un exemple de données longues à générer.

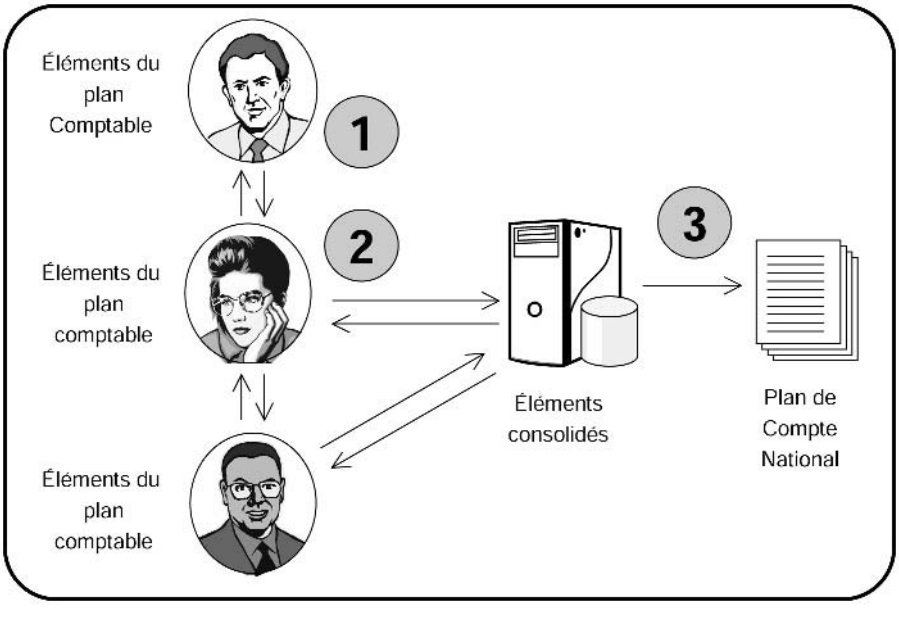

Figure 2.6 — Exemple de données longues à générer

- **Description**
  - (1) La saisie est effectuée dans Documentum (une application de GED).
  - (2) En parallèle, les modifications sont transmises par courrier électronique aux administrateurs de l'application Consolidation qui les saisissent.
  - (3) Puis les modifications sont transmises à une autre application.
- **Problèmes** – Il apparaît un problème de productivité, le plan de compte étant long à constituer.
- **Causes** – Un compte comprend environ 25 attributs dont 15 sont saisis/validés par la direction comptabilité et 10 par la gestion. Un *workflow* d'acquisition n'a pas été mis en place et la procédure manuelle est inefficace.

2.4.4 Données incohérentes

La figure 2.7 illustre des incohérences entre données d'applications.

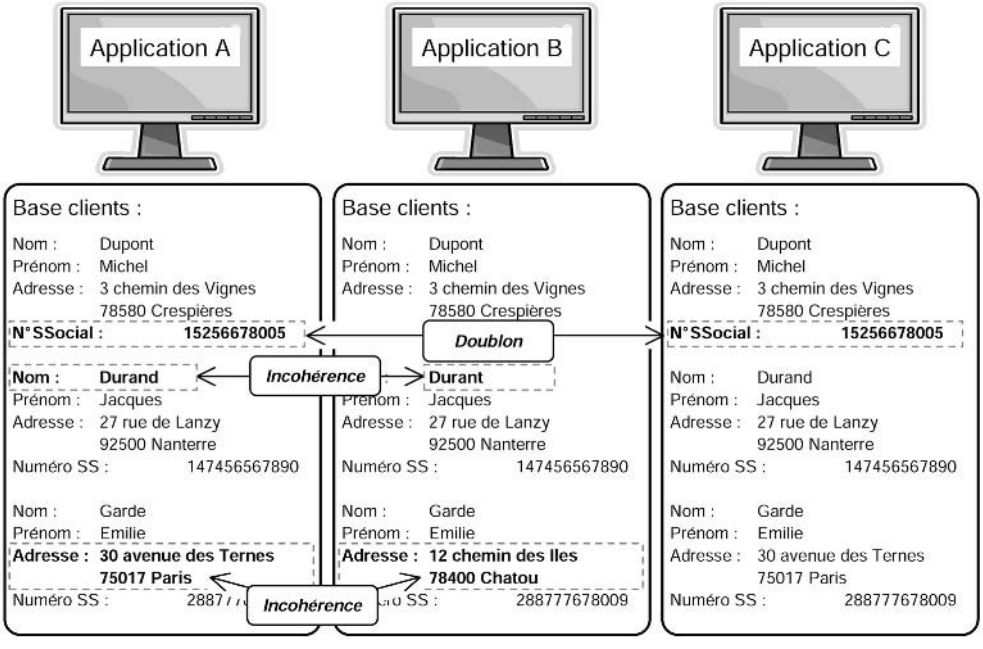

Figure 2.7 — Exemple de données incohérentes entre applications

Ces incohérences peuvent avoir plusieurs causes, qui peuvent être analysées de la façon suivante.

- **La collecte**
  - Pourquoi y a-t-il plusieurs producteurs de la donnée ? En effet, plus le nombre de producteurs est important, plus le risque d'erreurs et d'incohérence est grand.
  - Les périmètres des données fournies sont-ils disjoints ? Sinon, cela veut dire qu'on saisit plusieurs fois exactement la même donnée, il n'y a donc aucune valeur ajoutée.
- **Les traitements**
  - Où se trouvent les dégradations de qualité les plus importantes ?
  - Est-il normal qu'une application reçoive la même donnée de deux applications différentes ?
- **La mise à disposition de la donnée**
  - Pourquoi certains utilisateurs finaux disposent-ils de restitutions issues d'applications différentes ?

– Les valeurs des données restituées sont-elles toujours cohérentes dans les différentes restitutions ?

2.4.5 Autres exemples de problèmes fréquemment rencontrés

Outre les problèmes déjà évoqués dans les paragraphes précédents de ce chapitre et au paragraphe 1.3, on peut ajouter les exemples ci-après.

Manque de réactivité

La mise à disposition d'une nouvelle référence doit se faire rapidement.

Exemple : lors de l'établissement d'une facture, tout retard dû à des données manquantes (barème de prix par exemple) entraîne une perte de trésorerie.

Mises à jour trop longues et non simultanées

Toute modification des attributs d'une donnée de référence doit parvenir aux utilisateurs au moment précis de son application et de façon simultanée.

Exemple : la transmission tardive des nouvelles coordonnées bancaires d'un client peut se traduire par un impayé et engendrer les frais relatifs aux corrections, à la relance du client et à la perte de trésorerie consécutive.

Difficultés à ne diffuser que les données utiles

La mise à disposition des données de référence doit pouvoir être ciblée selon les structures locales pour n'offrir aux utilisateurs que des données utilisables.

Exemple : la mise à disposition de comptes de comptabilité à des unités non concernées pollue les autres métiers.

2.5 MÉTADONNÉES

Comme la qualité des données, les métadonnées sont un vecteur de valorisation des données.

Les métadonnées sont des données au sujet des données et de leur contexte. Quand on achète un produit, une étiquette donne des informations sur ce produit (composition, date, provenance, mode d'emploi, etc.). De la même manière, les métadonnées fournissent des informations nécessaires pour le suivi, le traitement, l'historisation ou le stockage d'une donnée.

Pour un document, les notices contiennent des informations sur la source du document (titre, auteur, date, sujet, éditeur…), la nature du document, son contenu informationnel (descripteurs, mots-clés, résumé). Pour un document numérique, ces notices s'appellent des métadonnées et sont contenues en général dans le document lui-même. Dans les bases de données relationnelles, les métadonnées incluent le

nom de chaque table et le type de chaque colonne dans la table (dictionnaire de données).

Les métadonnées permettent aussi de décrire les données utilisées dans les analyses et prises de décisions car elles incluent par exemple :

- la définition exacte des données (sémantique) ;
- la source des données (date, origine) ;
- la façon dont elles sont calculées ou agrégées (règles de calcul) ;
- les règles métier qui s'y rapportent ;
- le processus d'extraction, de transformation et de chargement qui a été mis en œuvre.

Autant les métadonnées sont utiles pour partager un vocabulaire commun dans une communauté qui peut être très large, autant elles peuvent entraîner des pertes d'informations lors des transferts de fichiers si l'on ne se conforme pas strictement aux exigences de tenues de registres de métadonnées exprimées dans les normes officielles (ISO/CEI 11179).

Les métadonnées portant sur une donnée permettent à celle-ci de devenir réellement information, ce qui est illustré figure 2.8.

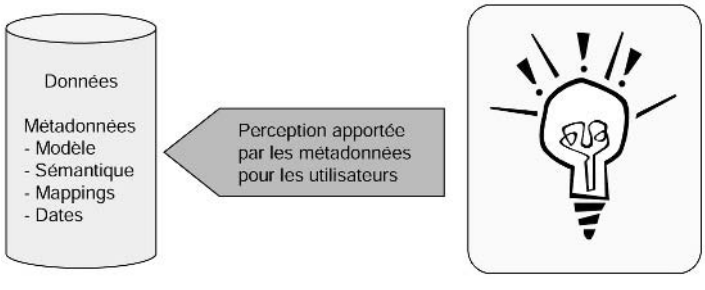

Figure 2.8 — Intérêt des métadonnées pour les utilisateurs

2.5.1 Types de métadonnées

Une distinction classique concerne les métadonnées métier et les métadonnées techniques.

- **Métadonnées métier** : il s'agit de décrire la référence et les correspondances entre les objets métier vus par un utilisateur (exemple : un lieu géographique) et les objets techniques correspondants (exemple : colonnes LIEU_ID et LIEU_DS de la table LU_LIEU). Ceci inclut les relations qui unissent les objets ainsi définis (typiquement des liens père-fils d'une hiérarchie).
- **Métadonnées techniques** : il s'agit d'une construction permettant de documenter et de maîtriser (notamment en termes d'analyse d'impact) les structures

manipulées dans les processus de chargement et de traitement des données (longueur d'un champ, définition d'un *mapping*…). Cela inclut les modèles de données.

Les métadonnées métier permettent :

- aux utilisateurs de comprendre la nature et l'origine de données dans leurs rapports ou analyses ;
- aux développeurs et utilisateurs clefs de mieux comprendre les données des référentiels afin de définir et parfaire la construction des mêmes rapports.

Les métadonnées techniques servent à :

- spécifier, maîtriser, et maintenir les référentiels de *business intelligence* (BI), les applications pour les alimenter, les calculs dans les rapports, et la qualité des données en général ;
- piloter les outils dans beaucoup de cas (par exemple on spécifie les transformations à l'outil ETL et il les exécute sans programmation au sens traditionnel) ;
- effectuer des analyses d'impact lors des modifications et évolutions des systèmes BI et de leurs sources ;
- tracer l'origine de données calculées en cas de doute (*data lineage*) ;
- comprendre les données des systèmes sources (analyse du code et des bases et fichiers de systèmes mal maîtrisés).

Pour les métadonnées métier, certaines fonctionnalités de référentiel de métadonnées sont disponibles dans les outils décisionnels utilisés pour l'analyse et le reporting, comme les « Univers » de BusinessObjects ou encore « SAS Metadata Server ». Cependant, ces référentiels seront peu intégrés et difficiles à maintenir par des utilisateurs métier.

Par nature, les métadonnées décrivent le contenu de la donnée, la structure ou son environnement. On peut donc distinguer trois types de métadonnées :

- Métadonnée descriptive (s'applique au contenu d'un objet) : elle est utilisée pour décrire les données et identifier les objets.
  Exemple : format, domaine de valeurs.
- Métadonnée structurelle (s'applique à un objet) : elle fournit les informations sur la structure interne des ressources et sur les liens de la ressource (hiérarchie, segmentation…).
  Exemple : modèle de donnée.
- Métadonnée administrative (s'applique à l'univers de l'objet) : elle est utilisée pour gérer et organiser les données. Elle permet la gestion de la donnée d'une collection de données.
  Exemple : date de dernière modification, auteur, *mapping*, ordonnancement.

2.5.2 Les apports des métadonnées à la valorisation des données

Tenter de couvrir l'ensemble des métadonnées du système d'information, même dans le périmètre des seules données de référence, n'est pas un objectif viable. Conceptuellement, la maîtrise de l'ensemble des métadonnées est imaginable, mais l'effort nécessaire à la création et encore plus au maintien d'un tel système est économiquement coûteux.

> Maîtriser les modèles, leurs transformations tout au long de la chaîne de la donnée et partager entre les métiers un vocabulaire cohérent sont déjà des objectifs ambitieux !

Dans le périmètre des données de référence, les apports des métadonnées se situent au niveau de :

- **La gouvernance** (voir cette notion dans la troisième partie du livre) :
  – par la garantie d'une définition précise des données et des règles associées afin de garantir l'alignement avec le métier et des métiers entre eux ;
  – par une description de l'écosystème de cette donnée afin de permettre une analyse d'impact en cas de modification d'un objet ou d'une règle.
- **L'administration de la preuve** (sécurité, conformité) :
  En réponse aux contraintes de conformité, la gestion de l'instance de donnée et la maîtrise de la cohérence de la donnée au long du cycle de vie et de la chaîne de la donnée s'appuie sur les métadonnées. Ainsi, la conservation du numéro de version de l'instance, du nom du dernier modificateur… est supportée par des métadonnées.
- **La gestion du patrimoine informationnel** (sécurité, qualité) :
  – pour sa protection, en permettant d'aligner le cadre sécuritaire en fonction de la sensibilité de la donnée décrite ;
  – pour sa valorisation, en prenant en compte de nouveaux indicateurs de pilotage reposant sur les dimensions de valorisation de l'information.

  Par exemple, identifier le tiers créateur de la donnée permet de contrôler que, sur l'ensemble des données créées, le ratio d'erreur de tel ou tel tiers reste en dessous d'une norme et d'en contrôler l'évolution.

En résumé

La donnée est une ressource, un actif de l'entreprise. Pour maximiser sa valeur, la donnée doit être de qualité. À défaut, l'activité d'une entreprise est perturbée au jour le jour, des dysfonctionnements apparaissent, l'insatisfaction de ses clients grandit, générant des pertes et dégradant son image. La qualité des données, c'est l'aptitude de l'ensemble des caractéristiques intrinsèques des données (unicité, exhaustivité, fraîcheur, disponibilité, cohérence fonctionnelle, cohérence technique) à satisfaire des exigences internes (pilotage, prise de décision…) et des exigences externes à l'organisation (réglementations…). Leur amélioration est une démarche continue : l'approche « processus » se propose d'éliminer les causes de non-qualité à la source, tandis que l'approche « nettoyage » se propose de « dépolluer » a posteriori des données en défaut.

La maîtrise des métadonnées (données sur les données) participent également à augmenter la valeur des données.

3

Données et processus

Objectif

Ce chapitre a pour objectif de montrer :

– l'importance de la donnée de référence en tant que composant structurant des processus métier ;
– la nécessité de mettre en place, indépendamment des processus métier, des procédures *ad hoc* de gestion du cycle de vie des données de référence dans le cadre de processus référentiels ;
– l'impact d'une telle approche par les données sur la démarche d'urbanisation du système d'information.

Après avoir défini et précisé, pour les données de référence, la notion de processus, nous décrirons les notions de cycle de vie (métier et technique) des données. De manière plus générale, nous situons la gestion des données dans le cadre des activités d'urbanisation du système d'information.

3.1 PROCESSUS

Le mot « processus » est un mot très utilisé et dans des acceptions diverses et variées, ce qui est source de confusion.

Pourtant ce mot évoque une action pratique. Toute entreprise est en effet organisée pour la production de valeur. Cette production se concrétise par la fourniture d'un *output* (que celui-ci soit désigné par les termes « produit », « service » ou « livrable ») et par la consommation d'*inputs*, la valeur produite (ou ajoutée) étant alors la différence entre la valeur des *outputs* et celle des *inputs*.

On peut calculer la valeur produite soit en considérant l'entreprise comme un tout, soit en la subdivisant en métiers qui produisent, chacun, une valeur spécifique. On peut même subdiviser encore plus finement, l'important étant qu'à chaque niveau de découpage on puisse associer un *output*, des *inputs*, et une valeur. Une fois cette subdivision effectuée et les valeurs que produit l'entreprise identifiées, se pose pour chacune d'elles la question suivante : « Comment produire cette valeur ? ».

Pour y répondre, il faut considérer les différentes étapes de la production, les activités des divers acteurs (contenu, enchaînement), les moyens mis en œuvre (papier, téléphone, ordinateur, outils, machines, biens intermédiaires), les données nécessaires (en consultation, mise à jour et création) et les *outputs* produits. Lorsque l'on a décrit tout cela, on a décrit un « **processus », enchaînement d'activités concourant à la production d'une valeur.**

La définition des processus s'effectue dans le cadre des études d'urbanisme des systèmes d'informations. Ces processus sont en relation avec la stratégie de l'entreprise (on parle de « principe d'alignement stratégique ») qui est la base de toute construction des processus, systèmes d'informations et systèmes informatiques (figure 3.1). Les processus représentent l'organisation que l'entreprise utilise pour mettre en œuvre sa stratégie et atteindre ses objectifs.

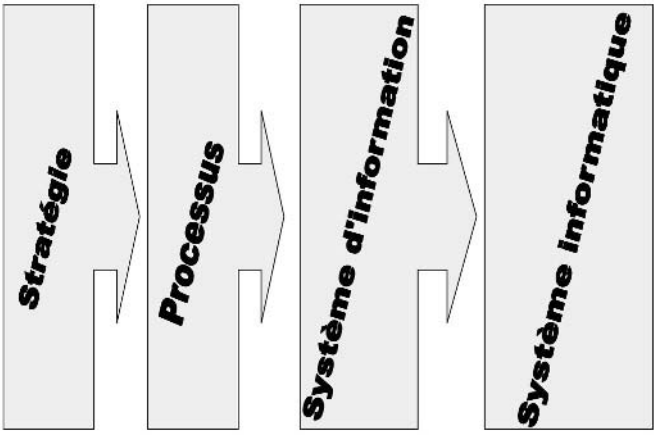

Figure 3.1 – Construction des processus

Un processus se décrit sous la forme d'un graphe. Les nœuds représentent les tâches élémentaires (**activités**). Un processus est en général déclenché par un *événement extérieur* (demande de devis, réception d'une commande, d'une lettre de réclamation, franchissement du délai de maintenance d'un équipement…) auquel il répond par une *action sur l'extérieur* (facturation, livraison, lettre, opération de maintenance…).

La figure 3.2 montre un exemple de processus.

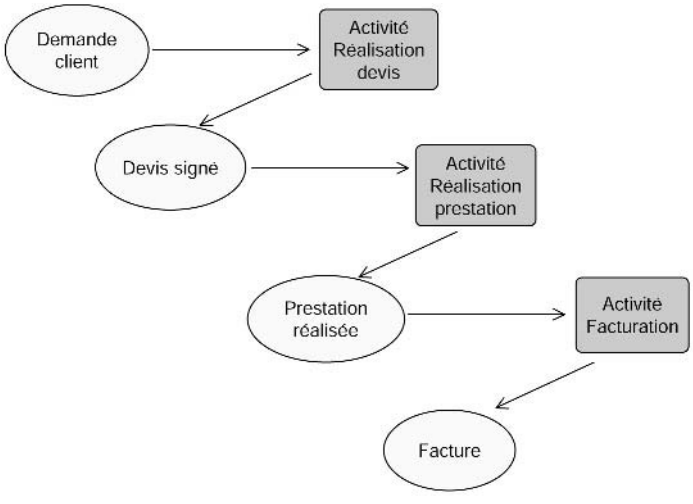

Figure 3.2 – Exemple de processus

3.2 PROCESSUS MÉTIER ET PROCESSUS RÉFÉRENTIELS

Historiquement, les systèmes d'information se sont construits en grande partie autour de domaines fonctionnels et des processus associés. Les processus métier et les données de référence sont au cœur des systèmes d'information depuis leur origine.

Cependant, l'utilisation de progiciels pour construire les SI a eu pour effet de favoriser l'outillage des processus, faisant ainsi gagner les métiers en autonomie et en réactivité face au marché. Mais cette prééminence des traitements **a eu pour effet pervers de minimiser l'importance de la donnée en tant que composant structurant de ces mêmes processus.** Les progiciels se sont par ailleurs arbitrairement approprié les données, ce qui les a rendues, aussi bien dans leur définition, leur accessibilité ou leur forme, dépendantes des processus ou des dits progiciels. Ainsi, le partage, l'enrichissement ou la surveillance des données est devenu plus difficile et coûteux**.**

Pourtant, les données de référence, plus pérennes que les processus, ne constituent-elles pas une valeur patrimoniale essentielle de l'entreprise ? Si les processus ont pour finalité de répondre à un besoin présent, a contrario, la donnée de référence gagne en valeur au fil du temps en fonction de plusieurs critères, notamment le nombre d'attributs détenus et leur qualité.

Assurer une forte qualité pour un grand nombre de données nécessite la mise en place de procédures permettant d'en contrôler l'acquisition, la gestion, les traitements mais aussi de garantir la pérennité de l'ensemble et la production des résultats attendus. Ainsi, indépendamment des processus métier, on pourra donc parler de **processus référentiels.**

Par convention, nous définissons donc (figure 3.3) :

- Le **cycle de vie métier** de la donnée (voir section suivante).
- Les **processus métier** qui peuvent couvrir tout ou partie du cycle de vie d'une donnée, même si en général un processus est à l'origine d'une et une seule transition. Par exemple, un processus de souscription peut faire passer un contrat successivement de l'état « souscrit » (signé) à l'état « inactif » puis à l'état « actif ». Plusieurs processus métier peuvent agir sur un même état de la donnée ou consommer cette donnée. Par exemple, un contrat d'assurance peut être résilié soit sur limite d'âge, soit pour cause de non-paiement des primes.
- Les **processus référentiels**, qui sont les processus spécifiques à la création, à la mise à jour ou à l'accès aux différents états du cycle de vie métier de la donnée.

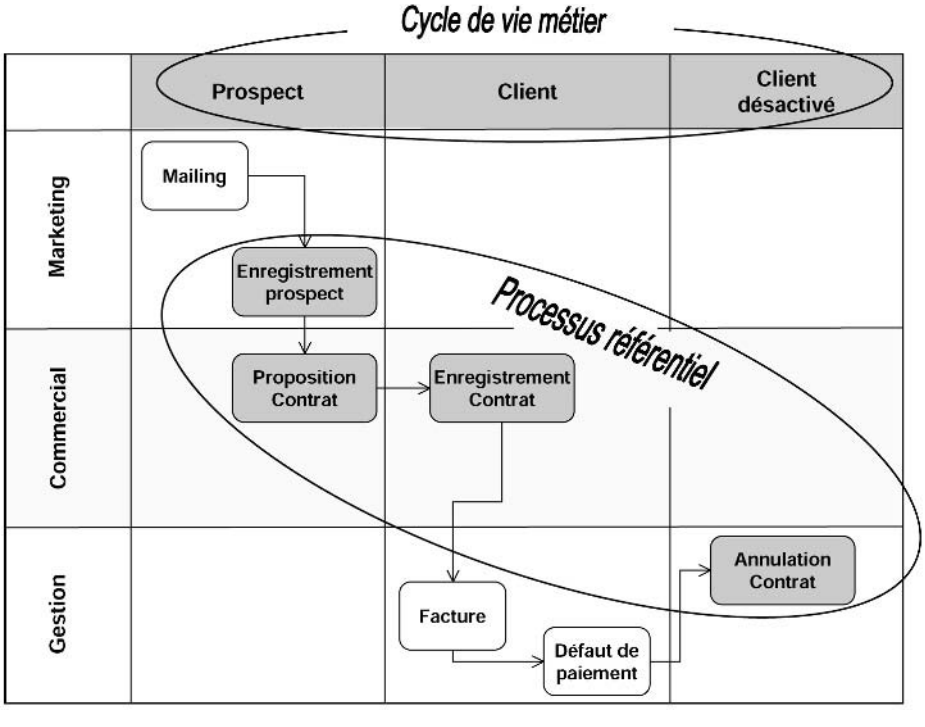

Figure 3.3 — Cycle de vie, processus métier et processus référentiel

À la figure 3.3, on a représenté le processus métier de gestion d'un client. Cela fait intervenir trois métiers de l'entreprise : le marketing, la vente et la gestion.

Le cycle de vie métier correspond à la séquence des différents « états » de la donnée client.

Le processus référentiel concerne les activités spécifiques liées à la création, à la modification ou à la suppression de l'objet client auquel on rattache un contrat.

> On note que ce processus référentiel est un sous-ensemble du processus métier, mais qui a vocation à être plus pérenne.

La figure 3.4 représente un processus référentiel relatif à la création d'un nouveau compte. On définit un *workflow* incluant la doctrine gestion et la doctrine comptable.

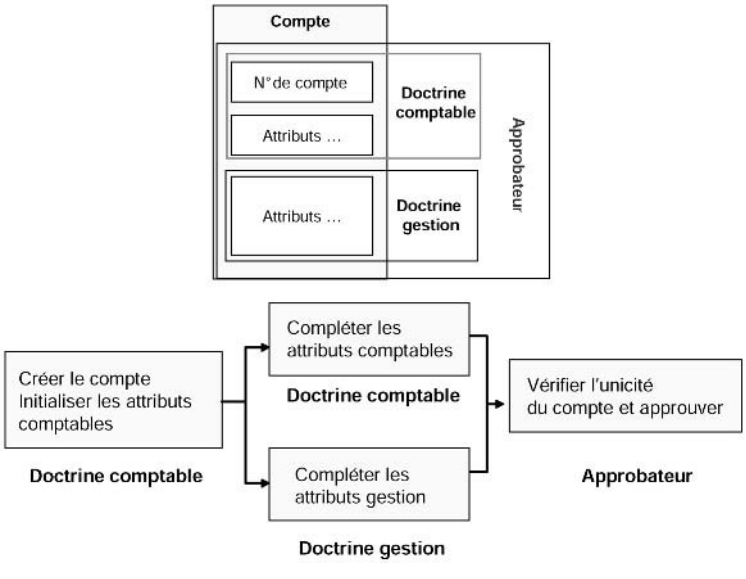

Figure 3.4 — Processus référentiel de création d'un compte

En pratique, décrire finement un processus référentiel est indispensable pour un projet de gestion des données de référence, car c'est ce processus qui sera implémenté.

3.3 CYCLE DE VIE MÉTIER

Le cycle de vie de la donnée doit être envisagé au-delà d'un simple point de vue applicatif (création, mise à jour, suppression).

Le cycle de vie métier correspond à la séquence des états « métier » d'une donnée. Le passage d'un état à un autre est délimité par un « événement métier ». On se situe ici au niveau d'un paradigme (exemple, le paradigme ébauché dans le paragraphe précédent, Client : prospect à demandeur à client actif à ancien client).

Il convient d'identifier les états que le référentiel doit couvrir car cela conditionne directement la taille et la complexité des projets. **Pour chaque état, les rôles et applications ayant droit sur la donnée doivent être définis**. Par exemple, pour l'objet client, si on traite un « prospect », la comptabilité n'a aucun droit sur l'objet mais est responsable de son périmètre quand l'état devient « client actif ».

En pratique, définir précisément le cycle de vie métier d'un objet est facultatif. Cela permettra d'associer un ou plusieurs services ou applications du système d'information à un objet métier dans un certain état (une gestion d'objet métier n'a pas de sens sans l'état associé). Cette démarche est donc souhaitable dans la définition d'une architecture SOA dans laquelle on précisera quels services sont disponibles pour gérer tel état d'un objet métier. Cela permet également de déterminer les attributs requis ou non pour chacun des états afin d'assurer la complétude de la donnée de référence pour chaque état. Enfin, comme évoqué, cela donne le moyen de préciser les droits liés à chaque état.

La figure 3.5 illustre un exemple de cycle de vie métier d'un produit.

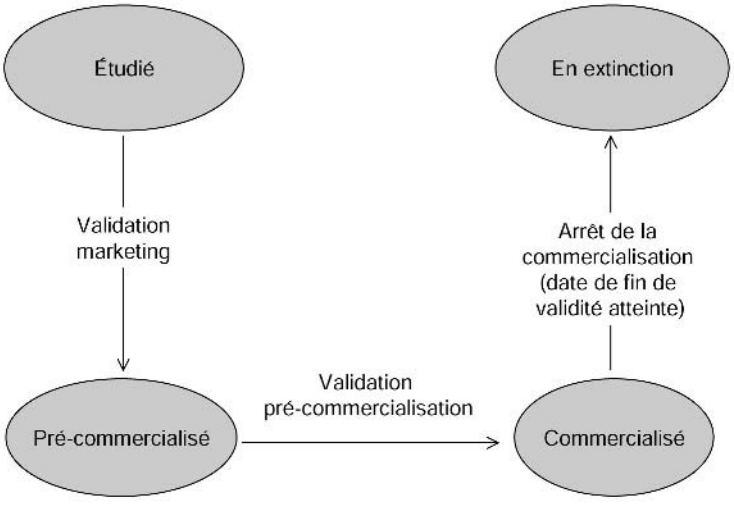

Figure 3.5 – Cycle de vie métier d'un produit

3.4 CYCLE DE VIE TECHNIQUE

On peut aussi décrire **un cycle de vie « technique »** de la donnée, lié à ces processus référentiels. Ce cycle de vie technique se prolonge au-delà des processus référentiels puisqu'il est aussi contraint par l'exploitation. Chacune des étapes décrites ci-après s'entend pour une donnée valide.

Création

Elle permet l'acquisition de la donnée.

On définit, a minima, les attributs essentiels autorisant la création d'un identifiant au sein du référentiel.

Mise à jour (modification)

La mise à jour permet la modification, la complétion (ou enrichissement) de la donnée afin de répondre aux besoins de l'organisation.

En fin d'étape, création ou modification, la donnée devient « valide » et donc consommable.

Ces étapes peuvent se terminer par une validation humaine en plus de la validation automatique.

Fusion

Cette étape est liée à la mise à jour. Elle peut intervenir indifféremment lors de la mise à jour ou après. Elle permet le rapprochement de deux données et leur dédoublonnage par création d'une nouvelle donnée.

Historisation

En fonction des besoins métier, l'historisation sera opérée à chaque mise à jour des valeurs de l'instance, à dates fixes ou lors d'un événement déterminé.

Les valeurs de l'instance sont alors enregistrées dans une base annexe. Tandis que le référentiel considère comme valides les dernières valeurs entrées de l'instance.

L'historisation est notamment utilisée dans les cas d'audit et de conformité aux exigences réglementaires.

Consommation

Cette étape assure la mise à disposition et/ou la diffusion de la donnée pour une utilisation au sein du système d'information (échanges, réplication, transferts, services web…). La donnée est utilisée au sein des applications métier.

Archivage

Une donnée peut-être archivée selon des règles définies par le métier lui-même lors d'une mise à jour ou à date fixe. En général, cet archivage donne lieu à suppression des instances à archiver dans la table de production et à leur création dans une table ou une base de données *ad hoc*. Par exemple, on conservera les trente derniers cours

de bourse d'une action dans la table de production (accès fréquents) et les cours antérieurs seront conservés soit dans une table annexe issue d'un partitionnement de la table active (table rarement accédée), soit dans une base de données archive, image de la base de production.

Suppression logique

La suppression logique considère la donnée comme supprimée, elle ne devient plus accessible à la majorité des utilisateurs ou aux applications. Son enregistrement est cependant conservé et peut encore être utilisé pour analyse ou pour des processus métier particuliers.

Suppression physique

L'information disparaît complètement du patrimoine de l'entreprise. Dans certains cas, cette suppression physique est soumise à certaines contraintes réglementaires (CNIL par exemple). Mais, de plus en plus, les données ne sont jamais supprimées physiquement mais archivées.

> Remarque : la sauvegarde n'est pas incluse comme étape. Nous considérons que la sauvegarde est essentiellement une fonction d'exploitation qui est mise en œuvre en réponse à un niveau de SLA (*Service Level Agreement*).

La figure 3.6 illustre ces différentes étapes.

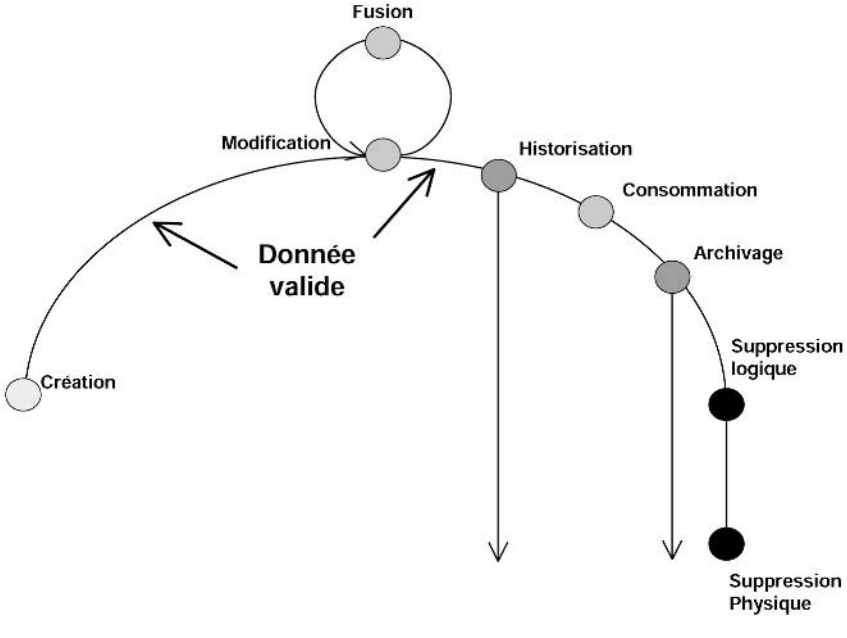

Figure 3.6 – Cycle de vie technique des données de référence

En pratique, à quoi sert ce cycle de vie technique ? Il sera utilisé lors des spécifications générales et détaillées des applications et/ou services associés.

3.5 URBANISME, URBANISATION ET DONNÉES

Nous définirons l'urbanisme du système d'information, par extension de la définition de l'urbanisme, comme l'ensemble des techniques et des méthodes permettant d'adapter le système d'information aux besoins de l'entreprise et de ses métiers.

L'urbanisme du système d'information est de plus en plus pratiqué dans les entreprises. Mais, si chaque entreprise conduit sa propre démarche, le point de convergence de toutes ces démarches est la prise de responsabilité des hommes de métier vis-à-vis des décisions concernant le SI de l'entreprise.

En effet, les entreprises ont maintenant conscience qu'au-delà de l'outil informatique, leur système d'information est devenu un enjeu fort et qu'à ce titre il doit être managé à part entière. Les entreprises devant sans cesse s'adapter au marché et à la concurrence, la réactivité et la souplesse (ou plasticité) de leur SI sont des atouts essentiels. La maîtrise de cette transformation constante de leur SI passe par une démarche d'urbanisme.

On utilise aussi le terme d'*urbanisation* plutôt que celui d'*urbanisme* pour mettre l'accent sur le travail progressif nécessaire pour **faire évoluer le système d'information vers une cible correctement urbanisée** et qui a été définie dans un plan d'urbanisme.

Urbaniser, c'est donc organiser la transformation progressive et continue du système d'information pour le simplifier, en optimiser sa valeur ajoutée et le rendre plus réactif et flexible vis-à-vis des évolutions de l'entreprise (stratégie, organisation) tout en s'appuyant sur les opportunités technologiques du marché. La figure 3.7 illustre cette démarche.

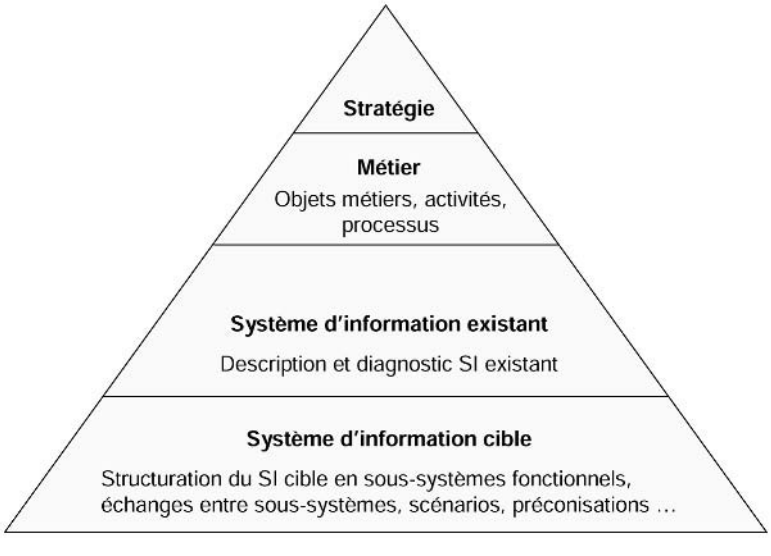

Figure 3.7 — Démarche d'urbanisme

Dans les très grandes entreprises, on découpe le SI en sous-ensembles qui correspondent en général aux différents métiers et donc, à de grandes directions (en lien avec l'organisation de l'entreprise). Chaque sous-ensemble est alors considéré comme un SI autonome du point de vue de son fonctionnement et de son évolution (le plus étanche possible avec les autres sous-ensembles). Il fait l'objet d'une démarche d'urbanisation propre.

Typiquement, on distingue quatre niveaux d'analyse (ou de vue) :

1 – La vue stratégique, constituée par la description :
– des objectifs de l'entreprise ;
– des objectifs du système d'information à urbaniser ;
– de la mise en correspondance entre ces deux sortes d'objectifs : il s'agit de « l'alignement stratégique ».

2 – Le système *métier*, constitué de l'ensemble des métiers, des processus de l'entreprise et des organisations qui y concourent, des activités et objets métier.

3 – Le système d'*information* (SI), constitué de l'ensemble :
– des objets métier spécifiés sous forme d'objets informatiques possédant des attributs, avec la sémantique associée (que l'on peut désigner par « information ») ;
– des fonctions regroupées en blocs fonctionnels appelés sous-systèmes (et qui préfigurent les applications) ;
– et des règles de gestion ;

utilisés par les métiers et les processus mis en œuvre par une même entité organisationnelle de l'entreprise.

4 – Le système *informatique*, constitué d'un ensemble structuré :
– de composants matériels ;
– de composants logiciels ;
– et de données ;

permettant d'automatiser tout ou partie d'un système d'information, et dont l'administration et l'exploitation sont assurées par une même entité organisationnelle (unité d'administration et d'exploitation).

Les figures 3.8 et 3.9 illustrent ce découpage.

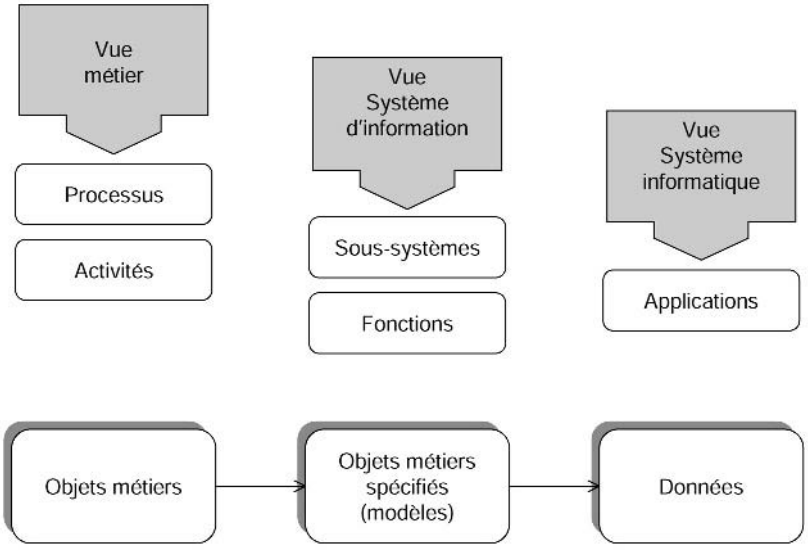

Figure 3.8 — Les trois vues d'une démarche d'urbanisme

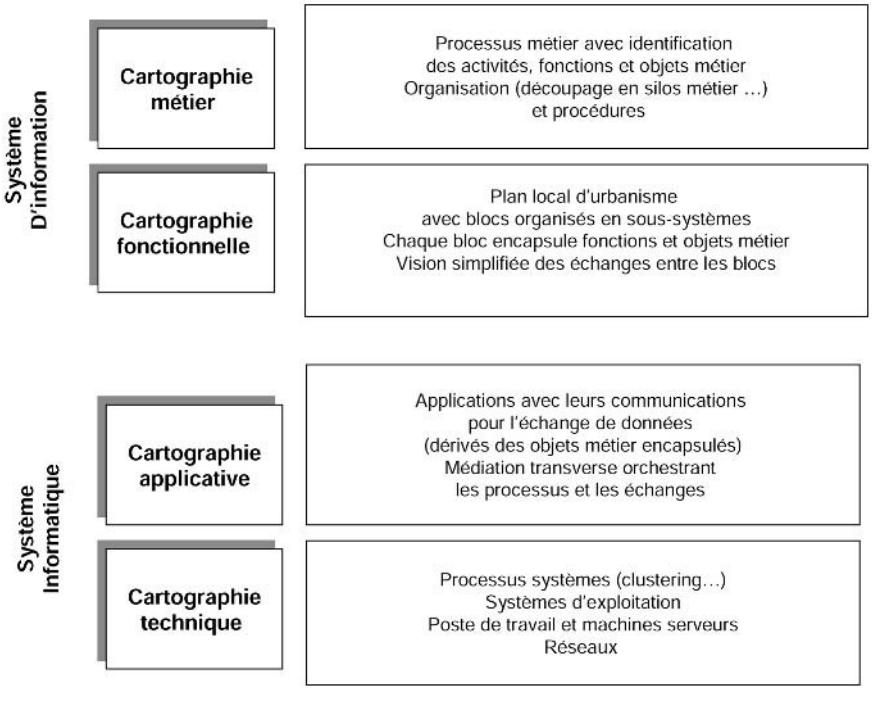

Figure 3.9 — Système d'information et système informatique

Dans la conception d'un système d'information, la *modélisation des données* est l'analyse et la conception de l'information contenue dans le système. On se reportera aux annexes pour plus de détails sur ce sujet.

Le modèle de données ne doit pas seulement définir la structure de données, mais aussi ce que les données veulent signifier (sémantique).

3.6 URBANISME ET DONNÉES EN PRATIQUE

L'urbanisme s'intéresse en priorité aux processus, en lien direct avec les métiers. Par exemple, le processus de commercialisation couvre (à très grosse maille) les activités marketing et vente.

C'est à ce stade que l'on identifie aussi les principaux objets métier manipulés par les activités et les processus, par exemple, les produits/services (conçus par le marketing) et les contrats (résultats des ventes). L'autre activité de l'urbanisme concerne le système d'information. Il décrit et analyse le SI existant (en termes d'applications et de flux échangés), et surtout le SI cible afin d'optimiser le processus métier décrit précédemment et la trajectoire de migration pour atteindre cette cible. Si nous reprenons notre exemple du processus de commercialisation, nous avons deux principales options dans le SI cible :

- un seul sous-système monolithique gérant à la fois le marketing et la vente, avec un seul SGBD gérant l'ensemble des données. Cette option ne va pas dans le sens de l'agilité souvent souhaitable dans les SI ;
- deux sous-systèmes indépendants, chacun ayant son SGBD, avec des échanges qui sont donc nécessaires entre les deux sous-systèmes.

Si l'on retient la deuxième option, on devra définir les événements déclencheurs des échanges, les documents échangés, le mode d'échange des données (message, fichier), le protocole d'échange, l'infrastructure d'échange adaptée à chaque type de document (outil de transfert de fichier, services web, EAI, ETL), les formats des documents et les *mappings* et autres transcodifications éventuels. On pourra aussi décider de créer un troisième sous-système chargé de gérer les données de référence indispensables aux autres sous-systèmes (qui contiendrait ici *a minima* le catalogue des offres, mais aussi les clients…).

Dans une approche descendante (*top down*), dès le niveau métier, l'urbanisme doit s'intéresser aux objets métier, et donc aux données de référence, au même titre qu'il s'intéresse aux processus métier. Logiquement, cette réflexion devrait même être menée en amont de la réflexion sur les processus. En effet, les objets métier représentent la base des métiers, sur laquelle s'appuient les processus, consommateurs et producteurs des informations portées par les objets métier. En pratique, les métiers ayant généralement plus de facilité à décrire leurs processus, l'urbaniste com-

mence donc son étude par cet axe. Ensuite, les processus permettent d'identifier des objets métier et d'amener la réflexion des métiers autour de ces objets.

Dans l'idéal, ces objets métier doivent être définis aussi précisément que possible : attributs, sémantique associée, relations. Dans une approche SOA (voir annexe sur la SOA), on identifiera également les opérations qui deviendront des services.

Dans une approche ascendante (*bottom up*), l'urbaniste décrit et analyse le système d'information existant (en termes d'applications et de flux échangés), le système d'information cible et la trajectoire de migration pour atteindre cette cible. À travers les flux, l'urbaniste peut alors identifier des objets métier.

Dans tous les cas, au-delà de l'identification des objets métier, il est important que, pour chacun d'eux, l'urbanisme puis les autres acteurs du SI, définissent la sémantique, un propriétaire, et décrivent l'utilisation de l'objet métier par les processus métier, les règles métier applicables à l'objet métier… Ces éléments représenteront une base de travail importante au moment de l'identification des données de référence éligibles au MDM (multi-utilisateurs, données réparties dans plusieurs applications…) et, pour chacune de ces données, de l'identification du maître et des esclaves.

On voit donc bien que les données font partie intégrante de l'ingénierie des systèmes d'information. Elles doivent être prises en compte le plus en amont possible des projets à composante informatique pour mettre à disposition les données de l'entreprise et fournir l'aide nécessaire à la conception des nouvelles données. Cela reste nécessaire pendant tout le cycle de vie des systèmes d'information, notamment à l'occasion de ses évolutions. Les données et les activités sont indissociables et vouloir représenter l'entreprise par l'une ou l'autre seulement de ces deux composantes n'est pas suffisant. Sans trancher sur le fait de savoir laquelle des deux est prééminente par rapport à l'autre, nous dirons simplement que les activités justifient les données et leur donnent du sens : sans activités il n'y a pas de données et inversement. **Par conséquent, il est important de gérer au mieux ces données, et particulièrement les données de référence.**

Dans la décomposition du SI en sous-systèmes on pourra en particulier distinguer trois types de sous-systèmes (figure 3.10) :

- *de production* (ou opérationnels) : ils regroupent les fonctions nécessaires aux utilisateurs pour dérouler les processus opérationnels ;
- *de pilotage* (ou décisionnels) : ils exploitent des données sur le fonctionnement et les résultats des processus opérationnels ;
- *référentiels :* ils gèrent les données de référence transverses aux autres sous-systèmes, voire aux différents SI.

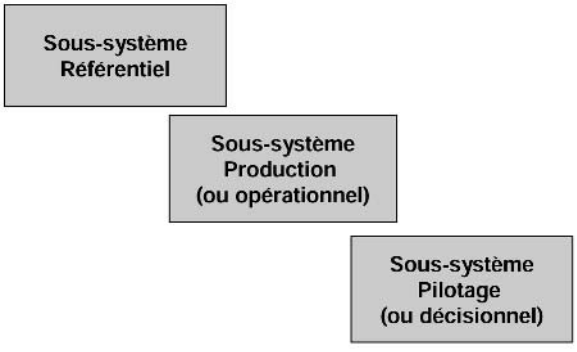

Figure 3.10 – Les trois principaux types de sous-systèmes

Quel que soit le découpage en sous-systèmes, il est souhaitable d'identifier si possible le **sous-système propriétaire d'un objet** qui, seul a des droits de création, de modification et de suppression sur l'objet (figure 3.11).

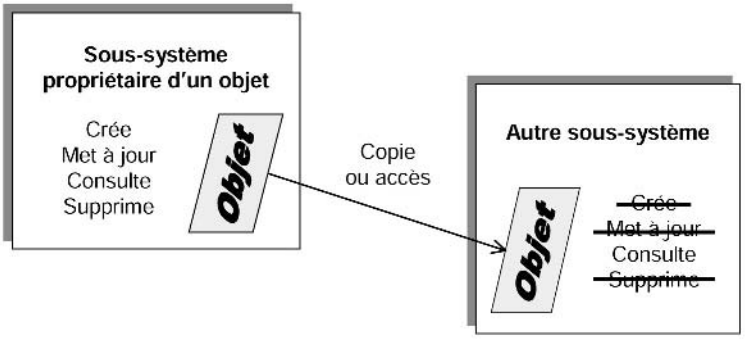

Figure 3.11 – Notion de sous-système propriétaire d'un objet

En résumé

Les données de référence, plus pérennes que les processus, se valorisent au fil du temps en fonction du nombre d'attributs détenus et de leur qualité. Mais assurer cette qualité nécessite la mise en place de procédures *ad hoc*. Ainsi il convient de définir, indépendamment des processus métier, des processus référentiels spécifiques qui traitent les données de référence tout au long de leur cycle de vie, métier ou technique. Il faut aussi déterminer le rôle de chaque application (ensemble des droits sur les états et changements d'état du cycle de vie) vis-à-vis de ces données. En particulier, il est essentiel d'identifier le moment du cycle, et donc l'application, où la donnée est reconnue valide et devient donc diffusable. L'urbanisme, par l'identification des objets métier, la définition de leur cycle de vie, la structuration du SI en sous-systèmes et l'étude de la couverture des processus tant métier que référentiels, doit faciliter cette démarche.

DEUXIÈME PARTIE

Mettre en œuvre : technologies et solutions

4

Typologies d'architectures

Objectif

Notre souhait, dans cet ouvrage, est de répondre au défi de la gestion des données de référence en donnant une approche simplifiée et progressive de l'outillage à mettre en œuvre. Notre approche repose sur quelques fondements théoriques concernant les architectures de gestion de données, à savoir :

– la distinction entre le « point d'acquisition » et le « point de vérité » de la donnée afin de définir un type d'architecture de base ;
– la déclinaison de ces architectures de base en « modes de déploiement » pour couvrir l'ensemble des cas possibles en entreprise.

Ce chapitre donne les clefs nécessaires pour résoudre tout type de difficulté dans la de gestion des données de référence.

4.1 FONDEMENT DES ARCHITECTURES

En termes d'architecture, la mise en œuvre d'un référentiel vise à **créer un point focal au sein du SI**. Ce point garantit la validité des informations détenues par le référentiel, c'est-à-dire leur niveau intrinsèque de qualité décrit précédemment. Dans la pratique une information valide devient disponible pour les processus consommateurs. On nommera ce point focal « **source de vérité** » **ou** « **point de vérité** »**.**

La position de ce point de vérité dans la chaîne de l'information et par conséquent dans le SI est d'une importance capitale dans la définition de l'architecture d'une solution de MDM (voir section 4.3).

Ce point focal se trouve entre **l'amont** et **l'aval** du référentiel[1].

Nous définissons **l'amont** par tous les éléments (IHM, applications, flux, processus, règles…) qui interviennent dans le processus transformant une donnée entrante en donnée valide (figure 4.1).

L'amont peut ou non faire partie intégrante d'une solution de type MDM, mais il est du ressort de l'architecte de considérer l'ensemble des briques applicatives et processus contribuant à l'acquisition des données du référentiel.

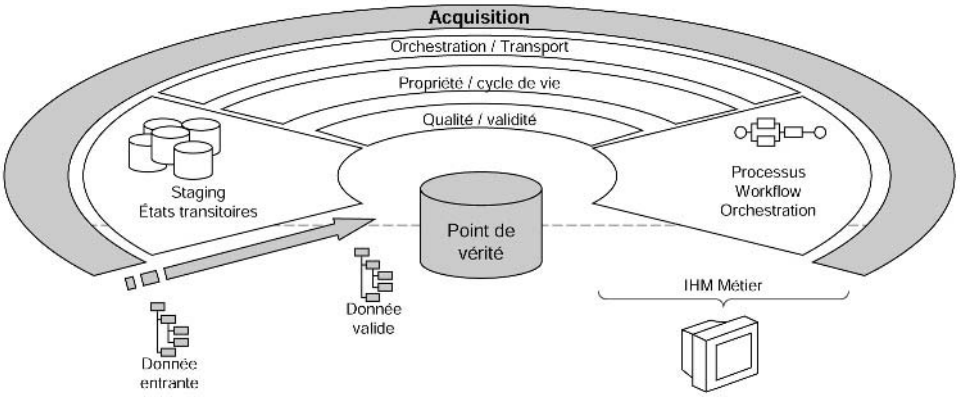

Figure 4.1 – L'amont de la gestion des données de référence

On nommera « **points d'acquisition** » les points d'entrée d'une donnée au sein du SI. C'est le point à partir duquel une donnée est saisie (saisie d'une adresse par exemple) ou à partir duquel une donnée externe au SI lui est intégrée (requête auprès de Dun & Bradstreet (base de données commerciale d'information), synchronisation en *Global Data Synchronisation* pour les SI de la grande distribution par exemple).

> On notera l'importance de ces points d'acquisition pour définir l'architecture des solutions de gestion des données de référence (MDM ou autre).

L'amont du référentiel est le lieu où sont situées les interfaces homme-machine. C'est aussi là que les règles de syntaxe ou de gestion sont vérifiées. De fait, l'amont du référentiel est plus orienté métier et fonctionnel que technique. C'est ce qu'il faut étudier en priorité afin de définir la typologie d'architecture.

Du point de vue du SI, l'amont est ce qui est compris entre le « point d'acquisition » et le « point de vérité ».

Du point de vue métier, l'amont peut bien entendu s'étendre au-delà du « point d'acquisition » car il peut couvrir des processus non outillés informatiquement.

1. Rob Karel, *Introducing Master Data Management*, Forrester Research, 10 novembre 2006.

Au contraire, **l'aval** du référentiel porte essentiellement sur des services techniques (figure 4.2).

L'aval englobe l'ensemble des éléments nécessaires à la mise à disposition de services de consultation et de diffusion des données et à la bonne intégration des données aux processus consommateurs (gestion événementielle, services de données, générateur de numéro d'instance, flux…). L'aval est important pour la mise en œuvre de l'architecture d'une solution, mais n'a que peu d'impact sur la définition de sa typologie.

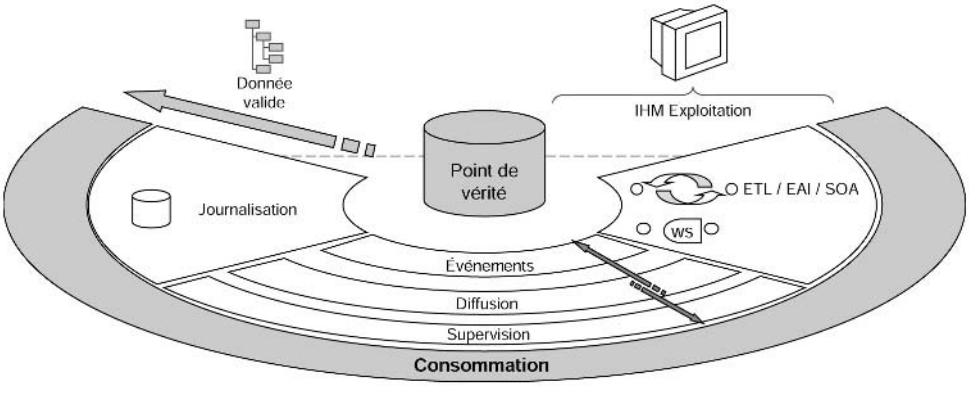

Figure 4.2 – L'aval de la gestion des données de référence

Le fondement de toute architecture de gestion de données de référence repose essentiellement sur la distinction et les rapports existants entre :

- le(s) point(s) d'acquisition de la donnée ;
- le point de vérité ou source de vérité (le référentiel en tant que tel).

La nature de la solution dépend donc de la distinction et du lien direct entre ces deux points et des liens de contournement éventuels existant autour de la solution (contraintes de synchronisation, voir la section 6.1.7).

La solution de gestion des données de référence est par nature un point focal pour les données qu'elle traite au sein du système d'information. Les processus métier « intervenant » (création, modification, suppression) ou « consommant » convergent donc vers la solution.

Schématiquement, l'architecture d'une solution de gestion de données de référence est composée de l'amont et de l'aval, séparés par le point de vérité (le référentiel des données valides en tant que tel). Cette notion de vérité/validité est elle-même polymorphe, car dépendante des états de la donnée, de son cycle de vie.

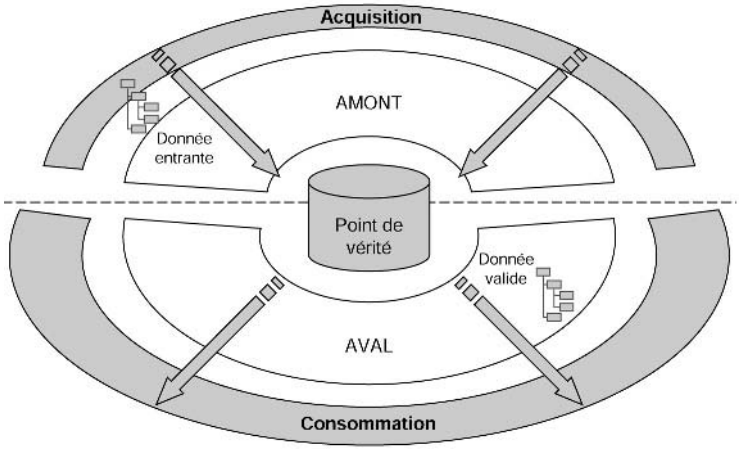

Figure 4.3 – Nature focale du point de vérité

Cette convergence architecturale reflète une orientation stratégique métier ou SI. Par exemple, focaliser son SI sur le référentiel client, c'est y faire converger l'ensemble des processus critiques de l'entreprise. Ce peut être la réponse architecturale à une stratégie d'entreprise. Une telle convergence s'opère généralement dans le temps et demande une mise en œuvre par étapes.

Plusieurs référentiels peuvent contribuer à une refonte du SI et à son accompagnement stratégique ou technologique. On veillera donc à industrialiser et à mutualiser les solutions mises en œuvre comme composantes de l'infrastructure. Cette mutualisation répondra pour les auteurs à la **notion de socle référentiel** (voir section 6.3).

On remarquera qu'une architecture convient à une donnée en particulier et que de multiples architectures peuvent convenir à différentes données **(on peut donc avoir une architecture différente selon la donnée traitée au sein d'une même solution).**

L'architecture des solutions de gestion des données de référence s'entend donc au niveau solution pour chacun des projets et en réponse à une problématique singulière. **Mais elle doit aussi s'envisager au niveau macroscopique du SI afin de répondre à des engagements dépassant le cadre du simple projet**. Le passage d'une échelle à l'autre s'effectue étape par étape en veillant à ce que la valeur d'ensemble ainsi créée surpasse la somme des valeurs individuelles des projets. La constitution du socle référentiel souligne cette génération de valeur au-delà d'un unique projet.

4.2 ARCHITECTURE ET CHAÎNE DE L'INFORMATION

Nous appelons **chaîne de l'information** l'ensemble des applications (maillons) au travers desquelles une donnée particulière circule : cette chaîne part d'un ou

plusieurs points d'acquisition et se développe jusqu'à son ultime application consommatrice.

En fonction de sa place *au sein de la chaîne de l'information*, la solution de gestion des données de référence répond à tel ou tel objectif. Cette place peut même évoluer dans le temps en fonction de l'évolution des objectifs et du niveau de rationalisation du SI. La solution référentielle vise aussi à mieux répondre à tel ou tel objectif en fonction de la place que l'on lui donne au sein de la chaîne de l'information.

Cette notion est d'ailleurs en relation avec l'identification des points d'acquisition et du point de vérité.

Nous proposons trois démarches selon la place du référentiel.

Référentiel en début de chaîne

Placé en **début de chaîne, le référentiel devient la seule autorité pour toutes les applications du SI.** Le référentiel est alors soit :

- l'unique point de saisie de la donnée ;
- la passerelle d'entrée d'un fournisseur de donnée externe (partenaire, sources de confiance, organismes d'état ou de normalisation…).

Les points d'acquisition et de vérité sont alors confondus.

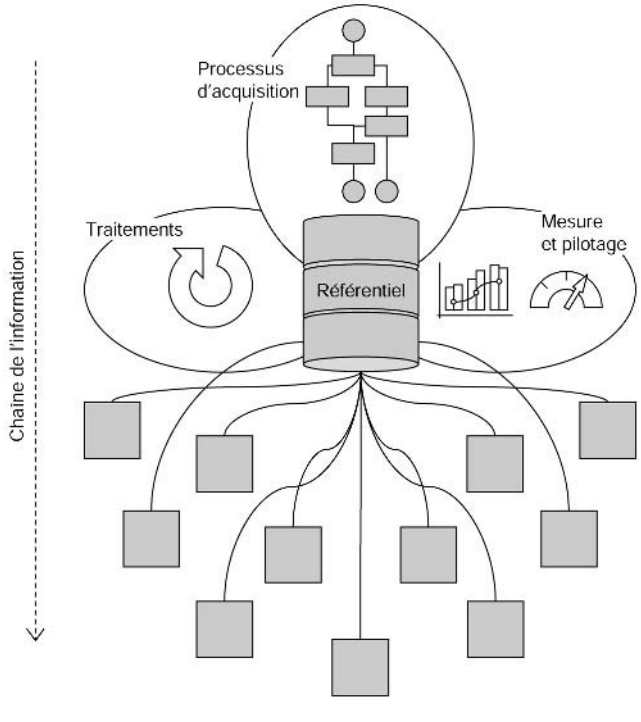

Figure 4.4 – Référentiel en début de chaîne

Un référentiel de début de chaîne assure le meilleur niveau de qualité possible pour les données de référence. Il maîtrise les processus d'acquisition et de mise à disposition de l'information au sein du SI.

Référentiel en milieu de chaîne

Placé en milieu de chaîne, le référentiel coopère avec les applications d'acquisition. Les points d'acquisition et le point de vérité sont distincts.

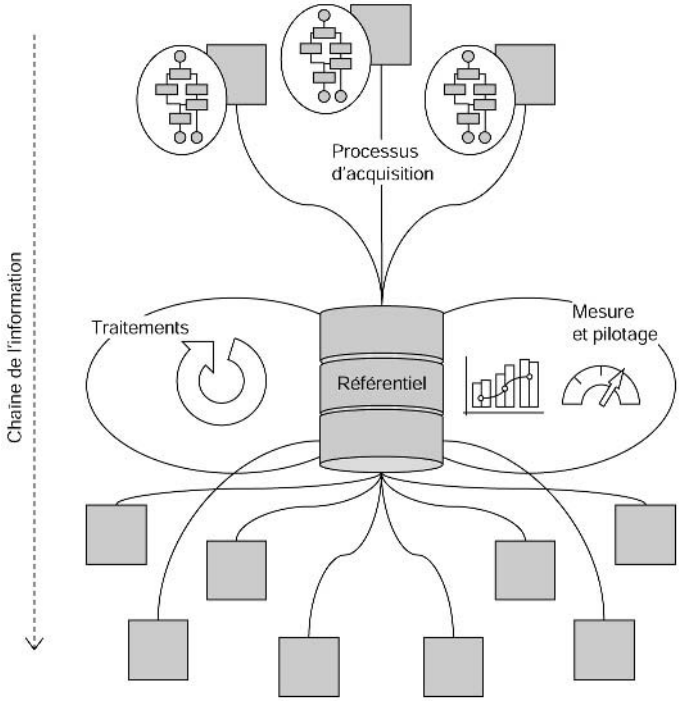

Figure 4.5 – Référentiel en milieu de chaîne

La solution de gestion de données de référence assure la redistribution de l'information et son contrôle qualitatif.

Référentiel en fin de chaîne

Placé en fin de chaîne, le référentiel est un réceptacle de données, utilisées par des applications consommatrices, indépendantes des applications sources.

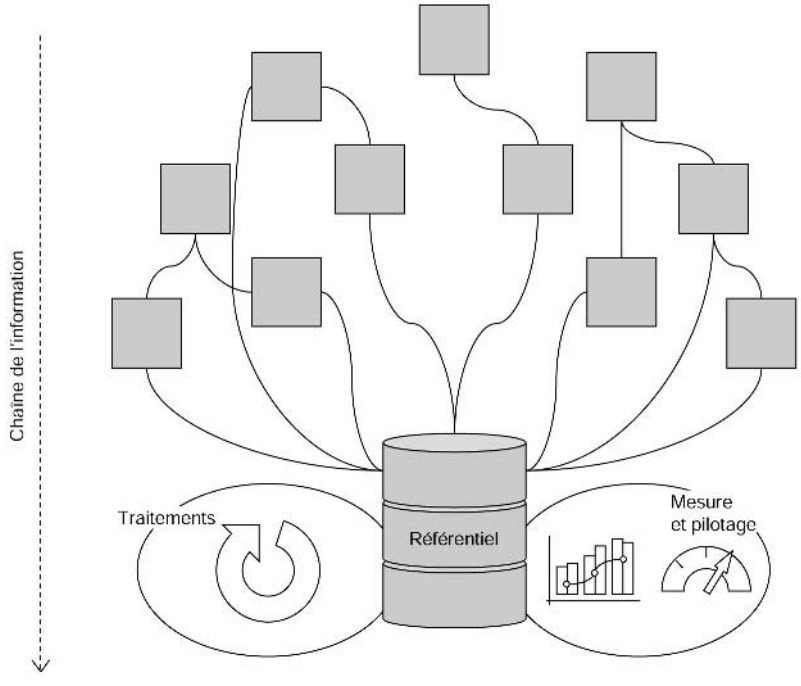

Figure 4.6 – Fin de chaîne

La solution de gestion de données de référence assure ici des traitements qualitatifs de redressement et de rapprochement. Ces traitements sont souvent complexes ou coûteux à mettre en œuvre pour des résultats moins opérants qu'en début ou milieu de chaîne.

La première question à laquelle nous devons répondre est celle du niveau de contrôle que nous désirons exercer sur les données. Cela permet d'identifier la position du référentiel dans la chaîne. Plus le référentiel est situé en tête de chaîne, meilleur est le contrôle sur les données. Il est ensuite possible d'en déduire le type d'architecture puis le mode d'implémentation.

4.3 LES QUATRE TYPES D'ARCHITECTURE POUR LA GESTION DE DONNÉES DE RÉFÉRENCE

Quatre types d'architecture peuvent être définis en fonction du niveau de liaison/adhérence entre le **point d'acquisition** et le **point de vérité**. Ces quatre types sont :

- consolidation ;
- coopération ;

- centralisation ;
- répertoire virtuel.

La figure 4.7 représente schématiquement ces quatre types d'architecture, afin d'identifier les points d'acquisition et de vérité.

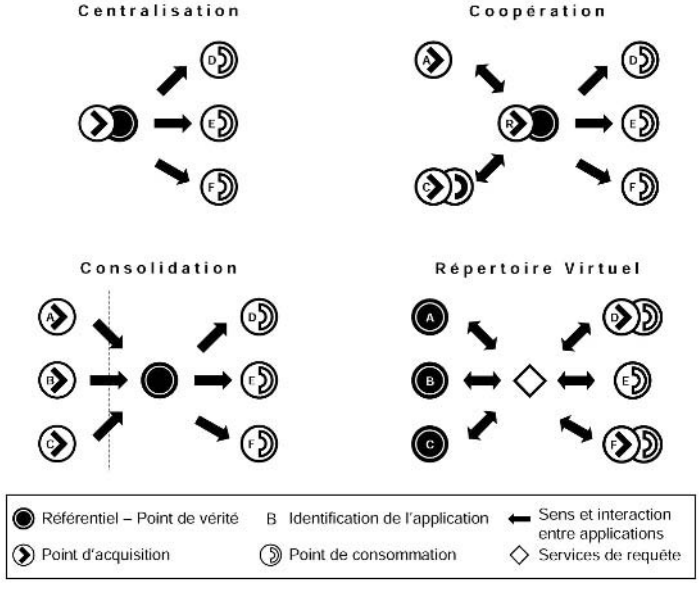

Figure 4.7 — Les quatre types d'architecture

La typologie des référentiels est en réalité, une généralisation. En effet, la distinction se fait réellement au niveau de chaque attribut de l'objet porté par le référentiel. Ainsi, au sein d'un même référentiel, on peut avoir 80 % des attributs correspondant au type centralisation, 15 % au type coopération et 5 % à celui de type consolidation. Parler de la typologie d'un référentiel, c'est en fait se référer au type majoritaire du référentiel. Rien n'empêche par exemple d'implémenter une solution dans laquelle une partie des attributs provient d'applications externes en coopération ou consolidation et une autre partie en saisie directe en centralisation.

Plus généralement, l'architecture à implémenter est à définir pour chaque donnée.

Cette approximation ne devra pas cacher au moment du projet les besoins fonctionnels induits par chaque type sur l'outil ainsi que les contraintes qui en découlent.

4.4 ARCHITECTURE DE CONSOLIDATION

Dans une architecture de consolidation, plusieurs sources de données alimentent le référentiel et les points d'acquisition sont distincts du point de vérité. Il n'y a pas de risque de désynchronisation transactionnelle entre les sources (points d'acquisition) et les consommateurs. Les sources de données sont indépendantes de la solution MDM et des processus et applications qui s'y alimentent. Au périmètre du SI concerné, les points d'acquisition de la donnée sont, dans cette architecture, les flux d'alimentation de la solution MDM.

Chaque type d'architecture est illustré par trois schémas : un premier schéma situe les points d'acquisition et le point de vérité, un second représente les données détenues par chaque niveau d'application, le troisième est une représentation simplifiée du contexte.

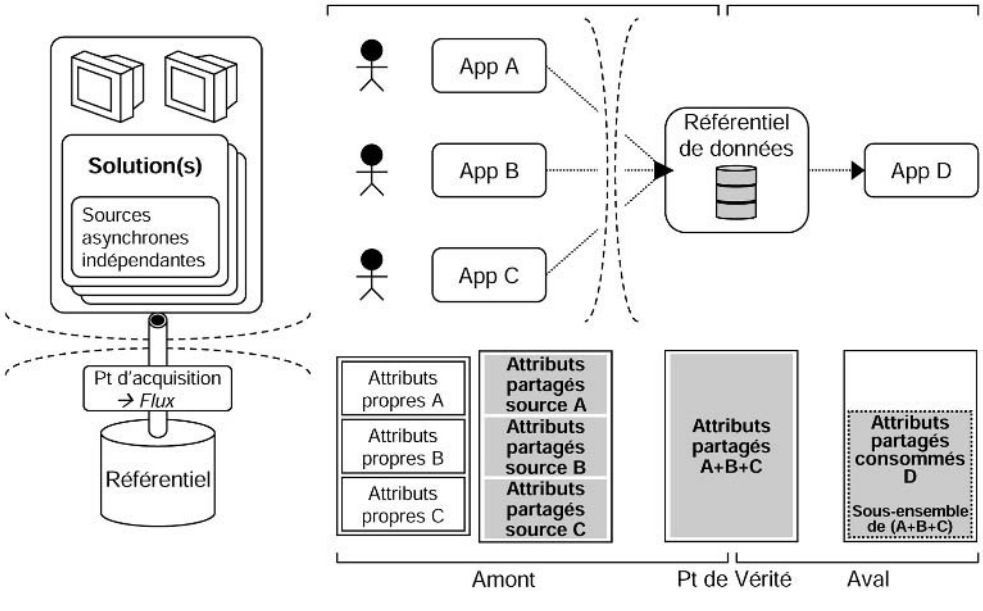

Figure 4.8 — Architecture de consolidation

Spécificité

Les liens entre les sources et le référentiel sont lâches. Ils présupposent une connaissance restreinte sur les données émises par les sources (contrat d'interface). La solution doit donc embarquer l'outillage nécessaire aux traitements de la qualité des données avant validation pour permettre d'atteindre les critères intrinsèques de qualité évoqués au chapitre 2 (unicité, complétude, exactitude, conformité, intégrité, cohérence).

L'architecture de consolidation demande, potentiellement et en complément du référentiel, des capacités de gestion qualitative de la donnée (DQM). Ce besoin est d'autant plus aigu que le référentiel traite d'un objet générant une forte volumétrie et que cet objet représente une réalité fortement volatile (par exemple, le référentiel client d'une grande entreprise peut comporter plusieurs millions de références et les données qui le composent sont rapidement périmées).

En revanche, les capacités d'intervention sur la donnée (interface de gestion, *workflows*) sont ici réduites à leur minimum. Au niveau des attributs, le référentiel peut agréger différents attributs partagés positionnés dans les applications source. Les applications consommatrices peuvent ne récupérer que quelques attributs du référentiel.

Type d'implémentation et avantages

Une architecture de consolidation est soit implémentée en début de chaîne d'information soit en fin de chaîne.

En début de chaîne, un référentiel de donnée valide une donnée entrante : on l'appellera « pré-référentiel ». C'est typiquement le mode d'implémentation destiné à valider un périmètre de données provenant de partenaires de l'entreprise. Il peut s'agir d'un pré-référentiel produit connecté en GDS (*Global Data Synchronisation*) ou à un portail fournisseur dans la grande distribution. En fin de chaîne, un référentiel de consolidation couvre l'ensemble des dimensions d'un objet métier. C'est généralement un référentiel destiné au décisionnel, le type d'implémentation est dit « référentiel analytique ». La solution mise en œuvre pour un tel référentiel opère des traitements qualitatifs a posteriori.

Exemples

L'architecture de consolidation est naturellement utilisée pour les projets de convergence ERP, en mode reprise. On l'utilise comme une architecture tactique avant de passer, par exemple, à la centralisation en mode production.

On note cependant que de tels traitements peuvent être complexes à mettre en œuvre pour des résultats parfois éloignés des objectifs visés.

Autre exemple, en fin de chaîne : une architecture consolidée de type « référentiel analytique » a comme objectif principal l'unification, qui tend à assurer la réconciliation des données et à consolider les attributs partagés par les applications décisionnelles.

4.5 ARCHITECTURE DE COOPÉRATION

Comme l'architecture précédente, l'architecture de coopération **utilise les applications existantes comme points d'acquisition de la donnée. Mais ces applications font partie intégrante de la solution référentielle amont** car :

- les processus de création et modification des données sont partagés entre ces applications et le référentiel ;
- elles sont dépendantes du référentiel car utilisatrices des données dont elles sont sources (préservation du référentiel comme source de vérité et protection contre les risques de désynchronisation des processus entre applications source et consommatrices).

Ainsi, chaque donnée saisie dans l'application source est transmise au référentiel puis le référentiel renvoie une validation à l'application source avant que celle-ci ne puisse utiliser la donnée saisie (figure 4.9, noter les doubles flèches entre source et référentiel qui correspondent au mécanisme de validation des données effectué dans le référentiel).

Au niveau des attributs, le référentiel de coopération peut soit agréger les attributs (harmonisation), soit recenser les attributs positionnés dans les applications source (système d'enregistrement distribué).

Rien n'empêche une application source sur un périmètre de données d'être aussi une application consommatrice sur un autre périmètre.

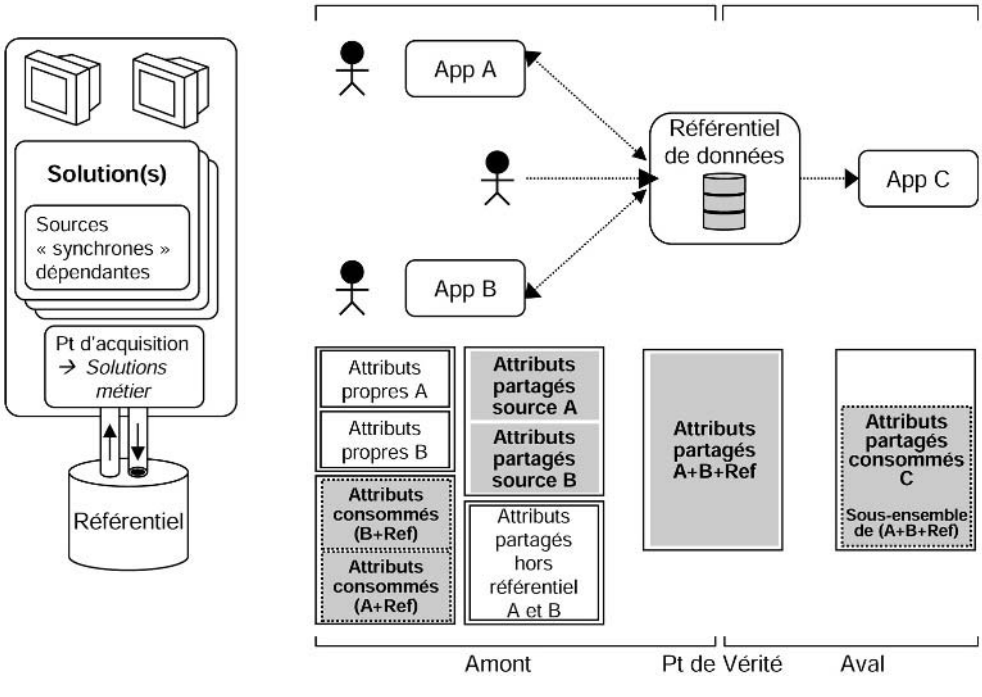

Figure 4.9 – Architecture de coopération

Spécificité

Cette architecture est la plus complexe à mettre en œuvre. Dans une logique de renforcement des contrôles de la donnée, elle peut être considérée comme transitoire tandis que dans une logique de transparence vis-à-vis des applications existantes, elle sera pérenne. Elle oblige à :

- une adhérence forte entre point d'acquisition et référentiel (a minima, elle demande la gestion d'un statut d'attente au sein de l'application source dans l'attente d'une confirmation de validation fonctionnelle par le référentiel. Le lien peut aller jusqu'à une confirmation synchrone) ;
- la solution doit composer avec l'ensemble des contraintes existantes sur les applications source.

L'architecture de coopération demande, en complément du référentiel, des capacités d'intermédiation renforcées afin d'assurer un lien sûr entre points d'acquisition et référentiel. Typiquement, elle sera implémentée de préférence sous la forme de services synchrones pour l'amont. Afin d'assurer le plus de liberté possible au métier, un mode dégradé doit être envisagé.

On notera que cette architecture demande une intervention sur les applications métier sources (drapeau d'attente de validation par exemple).

Enfin, cette architecture propose un défi intéressant du point de vue de la synchronisation des données à l'échelle du SI. Ainsi, des scénarios de synchronisation doivent être analysés. Ces scénarios se déclinent notamment en fonction du rôle des points d'acquisition (création ou complétion) et de la présence ou non de partage de données entre les applications amont. En effet, trois familles d'attributs peuvent être définies au niveau des applications source :

- les spécifiques ;
- ceux partagés entre applications et détenus par le référentiel ;
- ceux partagés entre applications et hors référentiel.

On préconise d'éviter le partage hors référentiel (tout attribut partagé est éligible au référentiel).

S'il n'y a pas d'attribut partagé hors référentiel, on favorisera le référentiel comme nœud de gestion événementiel (la modification d'un objet d'une application cible déclenche seulement un flux amont, la synchronisation SI est assurée par la gestion événementielle de la solution de gestion des données de référence).

S'il reste des attributs partagés hors référentiel, on gèrera indépendamment les événements et les flux des sphères partagées par le référentiel et celles partagées hors référentiel.

Type d'implémentation et avantages

L'assemblage de ces multiples applications implique un outillage spécifique pour le contrôle des erreurs et flux de traitements. Les outils d'intermédiation sont ici de

première importance. L'avantage de cette architecture est la préservation de l'existant.

L'expérience utilisateur est peu concernée par cette architecture. Le besoin en « gestion du changement » est ici minimum, les utilisateurs continuant à travailler au sein de leurs applications habituelles.

Exemples

Elle est particulièrement adaptée au sein d'un environnement progiciel complexe opérant sur un même processus métier (multiples instances, multiples applications).

C'est une architecture souvent employée dans les entreprises multicanaux pour synchroniser les informations client des frontaux (banque, assurance, télécommunication). Les solutions MDM de type CDI (*Customer Data Integration*) supportent généralement très bien ce type d'architecture.

4.6 ARCHITECTURE DE CENTRALISATION

Dans une architecture de centralisation, la solution de gestion des données de référence apporte un support direct des processus référentiels (création/modification/suppression des données) : le point d'acquisition et le point de vérité sont ainsi fusionnés au sein de la solution.

Ce type d'architecture demande une forte adaptabilité de la solution afin de ne pas faire de la centralisation une limite aux demandes des évolutions des métiers. Les attributs partagés sont positionnés et gérés directement dans le référentiel et les applications aval ne consomment que les attributs utiles, en fonction de leur contexte propre (figure 4.10).

Spécificité

Cette architecture est nativement orientée utilisateurs. Interfaces homme-machine (IHM) métier et/ou *workflows* sont parties intégrantes de la solution.

L'adaptabilité aux besoins métier est ici nécessaire autant techniquement que dans la méthodologie projet (accompagnement, besoin de formation des équipes qui devront utiliser la solution).

L'architecture de centralisation demande, en complément du référentiel, des capacités de *workflow* afin d'assurer la gestion des processus référentiels au sein de la solution.

Enfin, une intégration au travers des IHM peut rendre transparente cette architecture du point de vue des utilisateurs. Par exemple, lors de la mise en œuvre d'un ERP ou d'un CRM, si l'intégration est réalisée au travers d'un portail, il est possible d'utiliser directement la solution de gestion des données de référence pour les processus concernant les données de référence et de rendre ainsi au progiciel son simple rôle de consommateur (esclave).

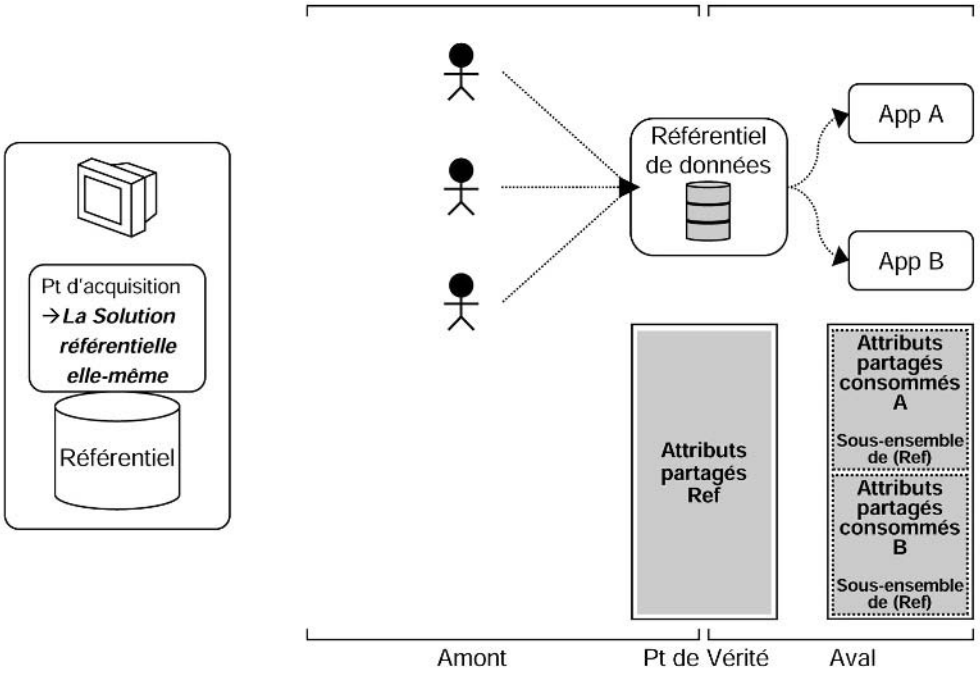

Figure 4.10 — Architecture de centralisation

Type d'implémentation et avantages

Cette architecture peut être considérée comme une **architecture cible à privilégier lors de la refonte d'un SI**. Cette architecture apporte le meilleur gage de qualité des données par une maîtrise complète des processus et règles appliquées et par l'asservissement de toutes les applications du SI consommatrices de la donnée supportée. Elle **autorise une gouvernance renforcée**, supportée au mieux par l'architecture de la solution, l'outillage technique de la gouvernance pouvant être plus facilement décliné et mis en œuvre.

Exemples

L'architecture de centralisation répond à un besoin de contrôle des processus d'acquisition et de modification des données référentielles ainsi qu'aux importants besoins en termes de gouvernance du référentiel. Elle est donc adaptée aux refontes en profondeur des grands processus métier.

4.7 ARCHITECTURE DE RÉPERTOIRE VIRTUEL

Une architecture de répertoire virtuel est une architecture correspondant, globalement, à une architecture de consolidation. Cependant, elle utilise une technologie

EII ou un EAI/ESB couplé à un référentiel en mode « système d'enregistrement distribué » afin de mettre l'information à disposition des applications consommatrices directement au travers d'un service de requête.

Remarque : si d'un point de vue technologique on peut imaginer que l'EII puisse être le vecteur de mise à jour des données (création, modification, suppression), une telle solution n'apporte aucune facilité de gestion ni de garantie du processus référentiel comme dans une architecture de centralisation. De même, dans le cadre d'une architecture en coopération, l'EII n'a aucun intérêt par rapport à un SI urbanisé reposant sur une solution de type EAI ou ESB. Cette architecture est donc décrite afin de couvrir l'ensemble des architectures, mais nos recommandations se portent sur la consolidation, la coopération et la centralisation. Ses principales caractéristiques sont les suivantes :

- les points d'acquisition sont les applications source de la donnée en elles-mêmes, mais elles doivent être décorrélées (en termes de processus) des applications consommatrices de la donnée ;
- **ce type d'architecture ne convient que pour des consommateurs ponctuels de la donnée**, sans exigence de gouvernance important.

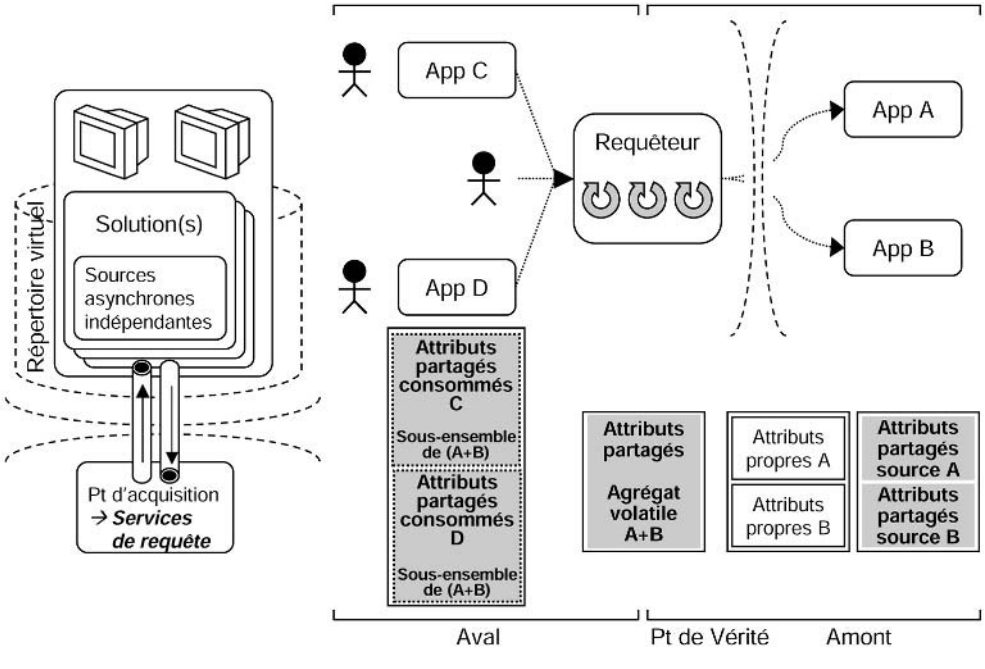

Figure 4.11 – Architecture de répertoire virtuel

Spécificité

Cette architecture est nativement orientée vers la consommation de la donnée. Elle correspond à des besoins ponctuels et spécifiques des métiers en termes de consommation, notamment dans le cadre d'une architecture décisionnelle. Elle n'est pas recommandée dans la mise en place d'une démarche de gestion des données de référence car l'outillage n'offre que peu de capacité de gouvernance.

L'architecture de répertoire virtuel est l'architecture type d'utilisation d'un EII. Elle peut nécessiter, en complément, des outils d'intermédiation en interaction avec le requêteur.

Type d'implémentation et avantages

Cette architecture peut être considérée comme une architecture ponctuelle, non pérenne :

- elle apporte une réponse à court terme à un besoin spécifique de consolidation ;
- l'investissement nécessaire pour sa mise en œuvre devra à chaque occasion être comparé aux apports d'une solution de gestion des données de référence, notamment dans le cadre d'une démarche généralisée et industrialisée.

Exemples

L'architecture de répertoire virtuel est adaptée à quelques solutions décisionnelles. Les EII sont, d'ailleurs, régulièrement intégrés aux outils de *reporting* des solutions décisionnelles ou aux ETL. Ils permettent de contourner la mise en œuvre de cubes spécifiques au sein du *datamart* (entrepôt de données spécifiques) pour les indicateurs peu usités et/ou mobilisant peu d'information.

On retrouve aussi les EII intégrés aux applications de portail. Ces solutions sont de moins en moins souvent vendues seules.

4.8 TABLEAU DE SYNTHÈSE ENTRE ARCHITECTURE ET CAS D'UTILISATION

Le tableau 4.1, ci-contre, classe les types d'architecture pour quelques cas d'utilisation classiques d'une solution de gestion de données de référence.

Tableau 4.1 - Synthèse architecture/objectifs

Objectifs	Consolidation	Coopération	Centralisation	Répertoire virtuel	Commentaires - Exemples
Améliorer la qualité et créer une vue unifiée des données par rapprochement (analytique, key mapping)	* * *	*	* *	- -	Référentiel de données et rapprochement de clefs pour utilisation par le décisionnel.
Obtenir une vision 360° (modification fréquente ou constitution du modèle par étape)	* *	* * *	* *	*	- Création d'un référentiel client rassemblant les informations des différents canaux commerciaux de l'entreprise. - Solutions EII possibles.
Supporter l'adaptation continue du modèle de donnée d'entreprise (modification fréquente ou constitution du modèle par étape)	-	* *	* * *	*	- Exemple : référentiel organisation. - Répertoire virtuel seulement utilisable pour la consommation.
Supporter une démarche d'urbanisation (EAI ou SOA)	-	* * *	* * *	*	Consolidation et répertoire virtuel seulement en consommation.
Réaliser un gestionnaire de paramètres	- -	-	* * *	-	Création d'un référentiel de tables et de listes de valeurs.
Refonte des Processus Métiers ou du Système d'Information	- -	* *	* * *	- -	Que ce soit en support d'un BPM ou en re-engineering du processus référentiel lui-même.

4.9 CONSÉQUENCES POUR LES MÉTIERS

Les conséquences sur les métiers diffèrent selon les architectures. En effet, sans pour autant identifier des critères discriminants quantifiables, on notera que plus on cherche à renforcer la maîtrise du processus référentiel, plus on est sensible à la donnée et à ses processus, et plus l'architecture est centralisée.

Au contraire, plus on tend vers une architecture centralisée, plus les objectifs deviennent ambitieux (processus de spécification, nombre d'acteurs à mobiliser, conduite du changement, nombre de systèmes concernés dans le SI). Cela implique un accompagnement important au changement (figure 4.12).

On remarquera néanmoins qu'un choix d'architecture n'est pas irrévocable et qu'on peut passer d'une architecture à une autre en suivant le même ordre que celui de la sensibilité au métier. Par exemple, on peut mettre en place un référentiel de consolidation en soutien d'une démarche BI, puis élargir l'utilisation de ce référentiel en le couplant au progiciel principal de création de la donnée (architecture de coopération). On pourra enfin refondre les processus référentiels en les faisant reposer directement sur le référentiel (architecture de centralisation) et en asservissant toutes les applications consommatrices.

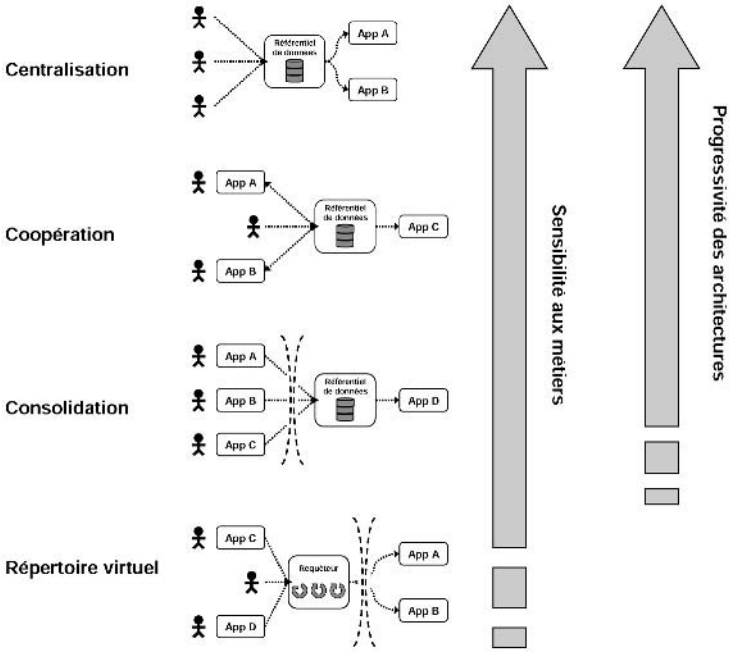

Figure 4.12 – Sensibilité aux métiers et choix d'architecture

4.10 SYNTHÈSE DES CRITÈRES DE CHOIX D'UNE ARCHITECTURE

La figure 4.13 résume les critères de choix d'une architecture.

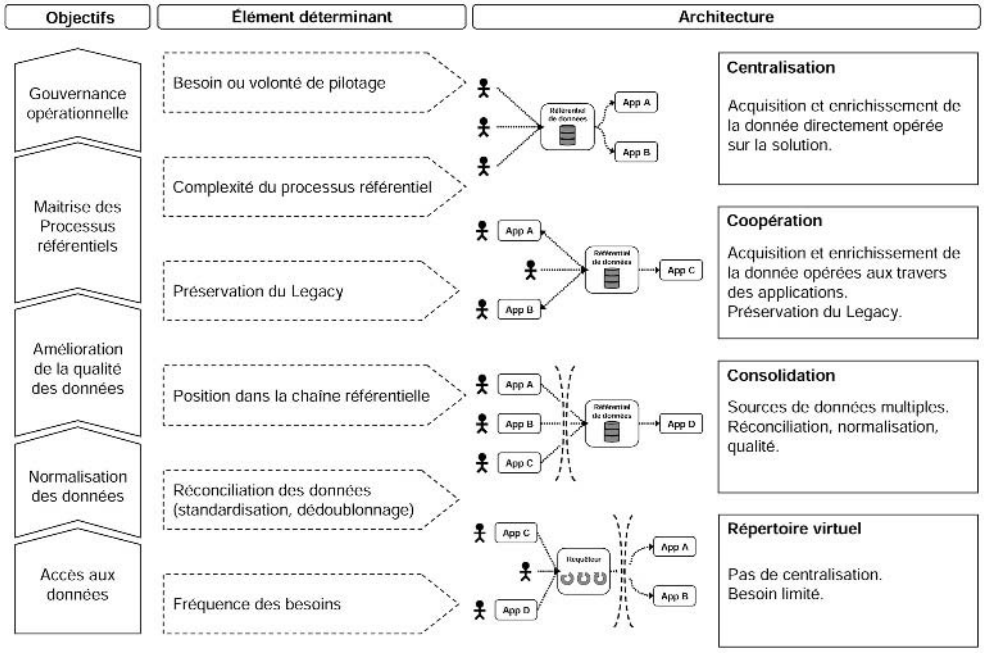

Figure 4.13 – Critères de choix d'une architecture

Dans une architecture d'entreprise, la solution référentielle doit se situer au plus haut de la chaîne de consommation de la donnée afin de fournir une information valide à l'ensemble du SI, et devenir le centre d'une couronne d'alimentation.

L'architecture de centralisation est, suivant cette règle, la seule à assurer une parfaite maîtrise de la donnée.

L'architecture de coopération est viable pour les données demandant une forte interaction extérieure et/ou possédant jusque-là des points de gestions dispersées.

Les architectures de consolidation et de répertoire virtuel sont, au mieux, des architectures tactiques.

4.11 COUVERTURE DU RÉFÉRENTIEL

Un autre choix influence la dimension et le cadrage des projets : quel est le périmètre des données éligibles ? Quelle est la couverture du référentiel ? Autrement dit : quel est l'objectif du référentiel et comment les données qu'il supporte y répondent ? En fonction de ces réponses, on obtient un périmètre de données très large ou au contraire très restreint. On peut positionner le référentiel au sein d'un spectre borné à ses extrémités par un référentiel globalisant ou synthétique.

Référentiel globalisant

Il comporte **des modèles qui rassemblent toutes les dimensions d'un concept au sein d'un même objet** (par exemple, le client considéré à la fois du point de vue marketing, vente, logistique, comptable…).

Référentiel synthétique

Il contient un **modèle minimum, reposant sur le « cœur de donnée » partagé par les entités du SI. Le « cœur de donnée » est ce qui permet l'identification dans le « monde réel » de l'objet représenté** (par exemple, le client est identifié par son cœur de donnée : nom, prénom, adresse, numéro de téléphone, e-mail).

La division proposée ici n'est pas exclusive. Le curseur entre « globalisant » et « synthétique » est généralement défini en cours de projet. Par exemple, on pourrait ajouter un point de livraison pour un distributeur en ajoutant un peu d'information logistique.

À l'échelle de l'entreprise, on peut imaginer un « réseau » de référentiels unifiés par un référentiel central d'identification, par essence purement synthétique. Mais on préférera généralement une démarche plus rationnelle et centralisatrice afin de passer d'une vision « technologique » des référentiels à une vision orientée « gouvernance » répondant mieux aux besoins métier.

Pour la suite, nous utiliserons la représentation symbolique de la figure 4.14.

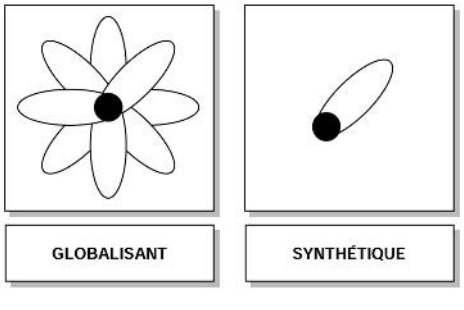

Figure 4.14 – Symboles des référentiels synthétiques et globalisants

4.12 LES MODES D'IMPLÉMENTATION DES RÉFÉRENTIELS

La figure 4.15 présente les modes d'implémentation des solutions de gestion des données de référence (qui peuvent être MDM, progiciel, développement spécifique) en se basant sur :

- **les types d'architecture :** s'ils se définissent au niveau de chaque attribut, on retient, au niveau de l'architecture du référentiel, le type le plus représentatif (par généralisation sur les attributs de l'objet) ;
- **la couverture des dimensions métier de la donnée** (globale ou synthétique) : elle est aussi sujette à interprétation et représente un choix de type d'implémentation, C'est un guide d'implémentation plutôt qu'une règle stricte.

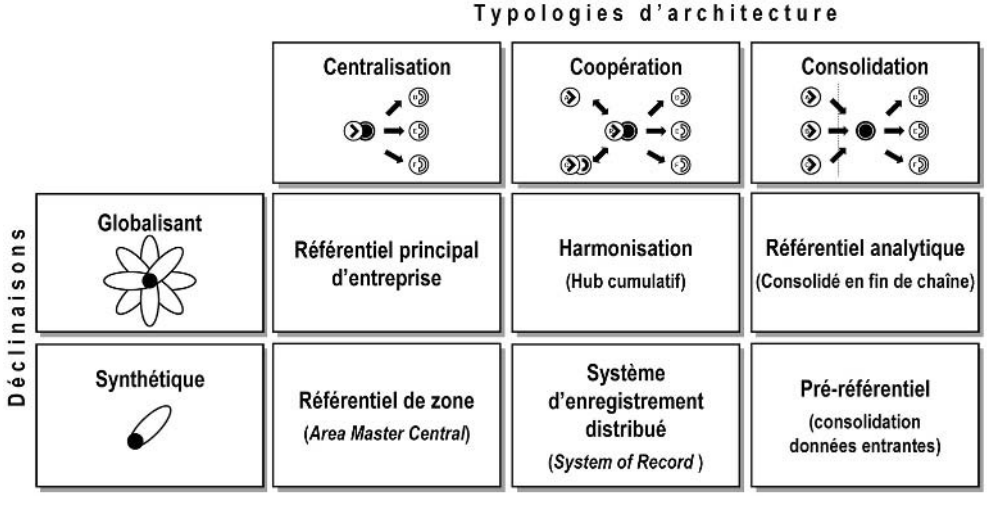

Figure 4.15 – Mode d'implémentations des solutions

On obtient six modes d'implémentation ayant chacun leurs particularités.

Référentiel principal d'entreprise

De type centralisation, c'est un référentiel qui a une large couverture métier. Il est mieux adapté aux structures centralisées qu'aux structures décentralisées. Il est par nature fortement collaboratif. Les processus d'acquisition des différents métiers, chaque étape de complétion de la donnée, sont outillés directement sur le référentiel ou au travers d'une intégration par IHM. Ce besoin collaboratif peut d'ailleurs excéder les capacités des outils MDM du marché et pourra être renforcé par un *workflow* (voir plus loin les architectures applicatives). C'est le type de référentiel le plus propice à la mise sous contrôle d'une donnée et à sa gouvernance.

Référentiel de zone

Le référentiel de zone est de type centralistion, en début de chaîne de l'information. Il permet la maîtrise d'une dimension métier, en particulier la gestion d'une donnée à l'échelle d'une direction, d'un service. Il est par nature collaboratif. La couverture étant moindre, des capacités collaboratives simples sont généralement suffisantes. Il permet la mise sous contrôle d'une donnée sensible pour tel ou tel métier.

Harmonisation ou hub cumulatif

Le référentiel d'harmonisation est de type coopération, situé en milieu de chaîne. Il reçoit, traite (normalisation, qualité, déduplication), enregistre et redistribue la donnée entre les applications frontales de l'entreprise (*front*) et les applications de support (*back-end*). Il est par nature transactionnel. Il doit resynchroniser le plus rapidement possible les applications frontales. Sa large couverture métier en fait le point de vérité pour l'ensemble des processus consommateurs. Il peut s'agir aussi d'une architecture de transition dans une logique de rationalisation des instances ERP d'une grande organisation.

Système d'enregistrement distribué

Le système d'enregistrement distribué est un référentiel de coopération, situé en milieu de chaîne. Il assure essentiellement l'unicité de l'information par le recueil des informations d'identification de la donnée enregistrées par les différents systèmes sources. Par ailleurs, ces derniers restent sources de vérité pour les données propres à leurs domaines respectifs. Il est par nature transactionnel. Il resynchronise et corrèle les identifiants entre les différents détenteurs d'information. Très axé sur la gestion des identifiants, c'est une implémentation, qui de notre point de vue, migrera rapidement vers celle du hub cumulatif.

Référentiel analytique

Le référentiel analytique est un référentiel de consolidation situé en fin de chaîne de l'information. Il permet une normalisation et le rapprochement des données issues des différentes applications de l'entreprise, ainsi que leur utilisation par le SI décisionnel. Il doit offrir des capacités de qualité (DQM) pour la normalisation et la déduplication ainsi que des capacités de *key mapping* (rapprochement de clefs ou de hiérarchies). Les traitements a posteriori que cela suppose n'offrent que rarement les gains escomptés. Dans ce cas aussi, se positionner plus haut dans la chaîne de l'information permet une meilleure maîtrise des données.

Pré-référentiel

Le pré-référentiel est un référentiel de consolidation situé en début de chaîne de l'information. C'est le réceptacle de données provenant de tiers (fournisseurs ou partenaire de l'entreprise). C'est une implémentation servant de sas entre deux SI. Les données sont réceptionnées, validées et normalisées avant d'entrer dans le SI de l'entreprise. C'est typiquement le mode d'implémentation des référentiels produits alimentés directement par les fournisseurs de la grande distribution avant l'enrichis-

sement par les services achats, marketing ou logistique. C'est aussi le mode d'implémentation préférentiel pour centraliser les données provenant de fournisseurs d'information à valeur ajoutée et fortement partagée (interaction avec Dun & Bradstreet par exemple).

Il offre des capacités de normalisation et des processus de validation (automatique et humain).

Le pré-référentiel n'apparaît pas toujours en tant que solution distincte. Un référentiel produit pourrait aussi bien couvrir les interactions avec les fournisseurs que la saisie des informations internes, le mode d'implémentation serait ainsi ambivalent entre référentiel principal et pré-référentiel.

Vocabulaire des éditeurs

On notera que les grands éditeurs du marché s'affrontent particulièrement sur la nature de leurs solutions et leurs modes d'implémentations. Ces éditeurs proposent notamment des visions telles que Central, Harmonisation, Consolidation pour SAP, ou Collaboratif, Opérationnel et Analytique pour IBM. Ces visions ne sont pas, pour nous, exclusives. Le lecteur remarquera que les acceptions des termes que nous utilisons dans l'ouvrage sont sensiblement différentes de celles de ces éditeurs.

Notre présentation propose une qualification reposant sur les besoins des organisations et non sur les capacités des outils. Les mises en œuvre doivent pouvoir évoluer d'une architecture type à l'autre, d'un mode d'implémentation à un autre sans modification importante de l'outillage.[1]

4.13 CHAÎNES RÉFÉRENTIELLES

Les besoins, les processus ou la complexité des organisations d'une grande entreprise peuvent entraîner la mise en œuvre de plusieurs instances de référentiels en charge d'un paradigme. Ainsi, plusieurs référentiels peuvent être chaînés, supportant ou non le même modèle physique.

Différents cas peuvent se présenter. Un exemple type est celui d'une chaîne référentielle supportant des interactions entre partenaires de l'entreprise et services internes. Ainsi, dans la grande distribution ou dans l'industrie, on peut trouver, en support du processus de référencement des articles, un pré-référentiel pour les partenaires externes accessibles en Extranet, un référentiel interne supportant l'enrichis-

1. Pour les solutions IBM, voir http://www-306.ibm.com/software/data/ips/products/masterdata/ et voir aussi en particulier ftp://ftp.software.ibm.com/software/fr/soa_summit/PDF_Archi/jminamdm_soa_summit_paris_jean_mina.pdf.
Pour les solutions SAP voir http://www.sap.com/france/solutions/netweaver/components/masterdata/index.epx
Concernant Oracle, voir http://www.oracle.com/master-data-management/index.html

sement des données, ainsi que des référentiels techniques locaux permettant l'irrigation en données des SI décentralisés.

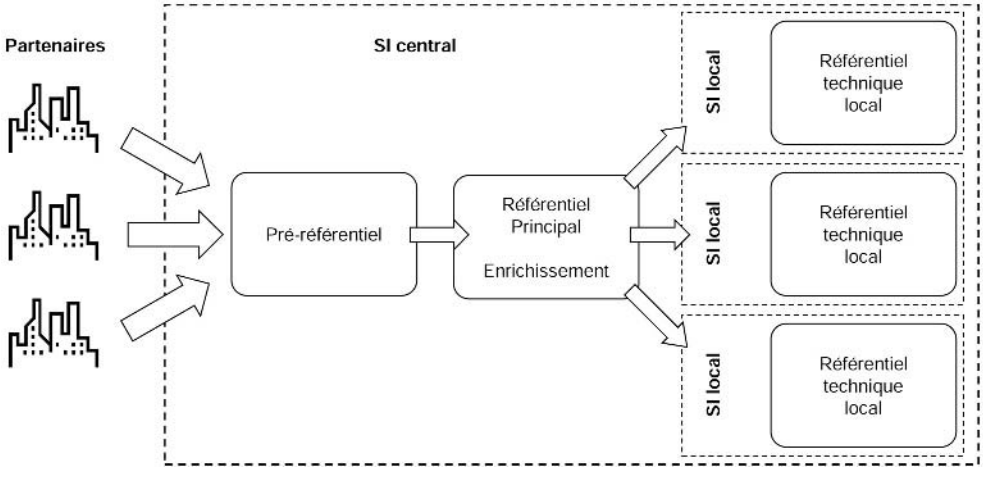

Figure 4.16 – Chaîne de référentiels

En résumé

La gestion des données de référence prend place entre des éléments d'acquisition en amont et des composants de diffusion en aval. Nous avons défini principalement trois types d'architecture (la quatrième, le répertoire virtuel, étant purement tactique) : centralisation, coopération et consolidation. Le choix de l'architecture se pose pour chaque donnée de référence, voire pour ses attributs. Il est important de définir les attributs qui seront effectivement gérés dans le référentiel : globalisant ou synthétique. En croisant les typologies d'architecture et la couverture attendue des domaines métier, on obtient six modes d'implémentation : référentiel principal d'entreprise ou référentiel de zone, hub cumulatif, système d'enregistrement distribué, référentiel analytique et enfin pré-référentiel.

5

Outillage d'une solution référentielle

Objectif

Ce chapitre présente un panorama des typologies de solutions envisageables dans le cadre d'un projet de gestion de données de référence.

Les logiciels affichant un tant soit peu de légitimité dans le cadre de la gestion des données sont nombreux et le choix devra être réalisé en analysant leurs apports en fonction des besoins.

Nous vous indiquons ici quelques éléments de choix à prendre en compte (grille d'analyse), le MDM étant analysé en détail dans le chapitre suivant.

5.1 TYPOLOGIE DES SOLUTIONS

Lors de l'étude de l'outillage des solutions de gestion de données de référence, les auteurs se sont trouvés face à de multiples choix quant aux applications à mettre en œuvre. En fonction du contexte, les solutions comparées sont soit des solutions *middleware* orientées données (MDM, DQM, EII, annuaire), soit des applications orientées métiers (PLM, CRM).

Vouloir outiller une solution représentant soit des objets (articles ou produits, par exemple) soit des tiers (employés, organisations, clients) influe directement sur les choix possibles.

MDM

MDM (*Master Data Management*) représente une suite de logiciels permettant le traitement des données destinées à qualifier et uniformiser le mode de description des informations pour en garantir une prise en compte correcte. Dans cette acception, MDM est une abréviation de « *MDM Repository* », à savoir une application référentielle en tant que telle.

DQM

Le DQM (*Data Quality Management*), ou gestion de la qualité des données, est une suite de logiciels qui permet de répondre à l'amélioration de la qualité des données de référence en assurant les fonctions de détection d'erreurs (typage, format, valeur, unicité) et d'amélioration de la qualité (correction, standardisation, complétion et dé-doublonnage).

EII

EII (*Enterprise Information Integration*) représente une catégorie de logiciels qui permet l'intégration, au sein d'une base virtuelle, des données disséminées dans les applications d'entreprise.

Annuaires

Ce sont des bases de données consolidant des attributs orientés personnes et organisations. Les annuaires implémentent généralement un protocole léger (LDAP) pour favoriser des réponses simples et rapides dans un cadre d'utilisation technique (par exemple l'authentification des utilisateurs).

CRM

Le CRM (*Customer Relationship Management*) est constitué d'une suite de logiciels ayant pour objectif l'amélioration des processus de gestion de chaque client en avant-vente, vente ou post-vente ainsi que d'un point de vue d'ensemble (étude et prévisionnel).

PLM

PLM (*Product Lifecycle Management*) représente une suite de logiciels ayant pour vocation la gestion des données et des processus relatifs à un produit. On y trouve des fonctions de collaboration pour la conception d'un produit, tout ce qui concerne son développement, ainsi que le contrôle qualité.

5.2 MDM (*MASTER DATA MANAGEMENT*)

Comme son nom l'indique, le MDM consiste à regrouper l'ensemble des données de référence de l'entreprise (*master data*) dans un référentiel standardisé qui joue le rôle de source d'alimentation lors de la mise à jour de tel ou tel système (point de vérité).

Concrètement, un outil MDM couvre le référentiel en tant que tel (*repository*), et offre des capacités de gouvernance de la donnée, d'acquisition, de validation et de diffusion. Contrairement à une simple base de donnée, un outil MDM permet la gestion et la gouvernance de la donnée. Alors qu'une base de données offre un modèle physique plat, cet outil offre plusieurs dimensions de gestion en fonction du cycle de vie, du rôle, du type de donnée… Comparer une base de données avec un outil MDM revient à comparer une multiplication avec une matrice.

Notons que ce type de solution peut être mono-référentiel, c'est-à-dire outiller une seule donnée, un seul paradigme, ou multi-référentiel, en outillant plusieurs, voire toutes les données de l'entreprise. Le référentiel contient donc un objet particulier ou l'ensemble des objets essentiels à la vie de l'entreprise (multi-référentiel) et décrit les liens qu'ils entretiennent entre eux : numéros de référence clients, fournisseurs, partenaires… Grâce à cette strate généralement associée à des mécanismes de contrôle et de validation, les objets sont modifiés de façon cohérente et les doublons évités. Au final, ce dispositif a pour but de garantir la qualité des données métier en phase de production.

La figure 5.1 positionne la solution MDM (fonctionnalités principales et échanges avec les autres applications).

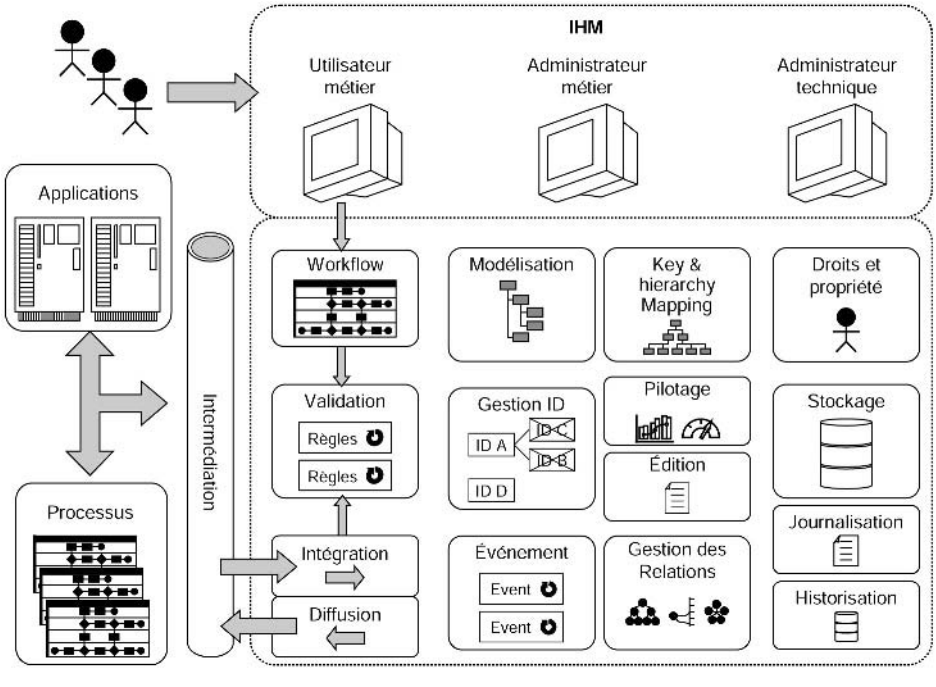

Figure 5.1 – Principe d'une solution MDM

Notons aussi que les différences d'organisation au sein des grandes entreprises et leurs divers besoins en gestion de données de référence peuvent nécessiter la mise en œuvre de plusieurs solutions basées sur différents outils de type MDM de différents éditeurs.

D'une manière générale, une solution de MDM :

- permet de définir des rôles et des droits d'accès individuels pour chaque étape du processus de gestion des données de référence et de son cycle de vie ;
- fournit des services de nettoyage de données pour comparer et dédoublonner les enregistrements (instances) ;
- offre des capacités de collaboration pour acquérir la donnée, la valider et coordonner les décisions de réconciliation et de rationalisation des données de référence ;
- propose des services de rapprochement sur clef ou de hiérarchies, de gestion des liens entre entités ;
- permet le pilotage du référentiel par le suivi d'indicateurs (par exemple, surveillance des données obsolètes ou du niveau de confiance – *decay & confident management* – ou encore des capacités d'auditabilité) ;
- prend en charge les événements, la détection des changements, la synchronisation bidirectionnelle et la réplication des données, afin de répercuter dans les systèmes concernés tout changement effectué dans le référentiel ;
- stocke et historise les données et leurs modèles ;
- est ouvert en termes d'évolution des modèles de données supportés.

Remarque : les échanges (« intermédiation » sur le schéma 5.1) s'effectuent avec des solutions d'intermédiation de type ETL, EAI, ESB, transfert de fichiers (FTP, CFT)…

Le fonctionnement d'un MDM induit donc, *a minima*, de :

- définir un responsable pour chaque catégorie de données de référence (client, produit, fournisseur, structure organisationnelle…). Ce responsable devient le garant de la qualité et de l'actualisation des données vis-à-vis de tous les systèmes, processus et personnes qui utilisent cette ressource partagée ;
- extraire des divers systèmes opérationnels, transactionnels et analytiques les données de référence de chaque domaine pour les charger dans des référentiels par domaine ou dans un *hub* central ;
- appliquer les normes de qualité des données pour obtenir un ensemble de données maîtrisées (en particulier en dé-doublonnant les enregistrements) ;
- définir les règles de réconciliation et de rationalisation des données de référence. L'objectif est d'obtenir pour chaque domaine une liste ou une hiérarchie optimale et compréhensible pour les utilisateurs, qu'il s'agisse d'individus ou d'applications ;

- définir les règles et modalités de synchronisation des référentiels ou *hub* avec les systèmes opérationnels et de *reporting*, afin de garantir que tous les systèmes utilisent, à tout moment, les bonnes données (même valeur, même version).

Une solution MDM tend donc à couvrir l'ensemble des besoins de gestion des données de références. Nous invitons le lecteur à avancer dans le chapitre suivant pour une description complète des fonctions d'un outil MDM.

5.3 DQM (*DATA QUALITY MANAGEMENT*)

La gestion de la qualité des données a pour objectif de garantir la fiabilité des données et, *via* des méthodes de contrôle continu, de permettre le suivi et le maintien de la qualité des données de l'entreprise.

Son positionnement en fait une brique indissociable des méthodologies MDM et *Business Intelligence*, et plus généralement, de tout processus visant à consolider et à harmoniser les données au sein du système d'information de l'entreprise. La motivation des entreprises pour la mise en œuvre de telles solutions est née de la prise de conscience qu'une mauvaise qualité des données génère des informations erronées pouvant entraîner des pertes financières importantes. Nous verrons que les outils de DQM sont aussi très utiles lors des phases de migration des données entre applications.

Comme nous l'avons décrit au chapitre 2, les dimensions clés (intrinsèques) d'une gestion de la qualité de la donnée sont :

- **Unicité** : existe-t-il de multiples et redondantes représentations des mêmes instances de données dans l'ensemble des instances détenues ?
- **Complétude** : toutes les informations requises sont-elles disponibles ?
- **Exactitude** : les objets de données représentent-ils bien les valeurs « du monde réel » qu'elles sont censées modéliser ?
- **Conformité** : l'information doit-elle se conformer à des formats standard ?
- **Intégrité** : les relations importantes entre objets sont-elles toutes présentes ?
- **Cohérence** : deux instances distinctes de données du même objet sous-jacent produisent-elles de l'information conflictuelle ?

Les domaines fonctionnels couverts sont :

- **Profilage :** génération de statistiques, analyse d'anomalie et évaluation.
- **Nettoyage et normalisation des données :**
  - conformité des valeurs aux domaines de restriction, aux constantes d'intégrité et aux règles métier ;
  - recomposition dans des dispositifs conformes aux standards industriels ou locaux, aux règles métier, et aux bases de connaissances.

- **Analyse de similarité et consolidation :** identification, liens et fusion des données similaires au sein des ensembles de données.
- **Enrichissement :** addition d'informations (par exemple, localisation géographique du client).
- **Surveillance** : mise en place des contrôles permanents assurant la conformité des données vis-à-vis des règles métier définissant la qualité des données de l'entreprise.[1]

5.3.1 Zoom sur l'évaluation des sources de données

Cette étape est souvent nommée « *profiling* » ou « *quality assesment* ». Elle repose sur une démarche maintenant classique :

- analyse des colonnes (indicateurs qualité pour la complétude des données) ;
- analyse des lignes (indicateurs statistiques de conformité) ;
- analyse des tables (détermine les dépendances fonctionnelles) ;
- analyse des clés primaires ;
- analyse croisée des tables (redondance des données) ;
- analyse des relations entre tables (clés étrangères ou secondaires) ;
- normalisation (prépare les données pour des analyses ultérieures facilitées).

Il est également nécessaire de confronter avec le métier :

- l'évaluation du modèle par la partie MOA du projet ;
- la complétion éventuelle des règles métier et du modèle de données par la MOA.

Notre présentation est largement inspirée des méthodes Informatica et IBM.

La figure 5.2 schématise les résultats obtenus à l'aide d'un outil DQM de *profiling*.

1. Voir notamment Informatica sur le sujet : http://www.informatica.com/products/data_quality/. Voir aussi IBM (depuis de rachat d'Ascential) : http://www.redbooks.ibm.com/abstracts/sg247508.html?Open et http://www.redbooks.ibm.com/abstracts/sg247546.html?Open

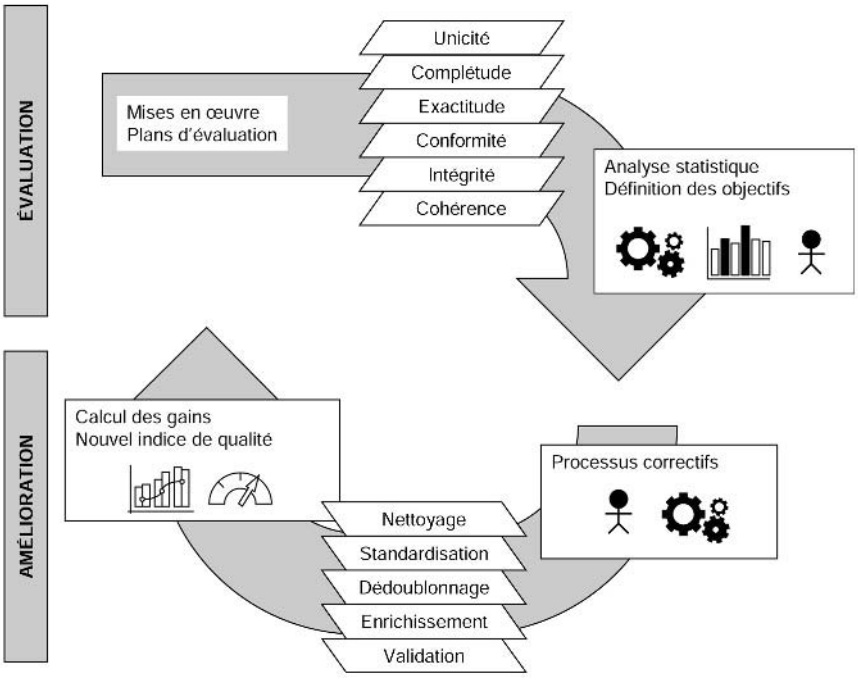

Figure 5.2 – Profiling et corrections des données

5.3.2 Zoom sur la qualité et la migration des données

La figure 5.3 explique comment améliorer la qualité des données **pour un projet incluant une migration de données**. Il s'agit d'un processus itératif d'évaluation et d'amélioration.

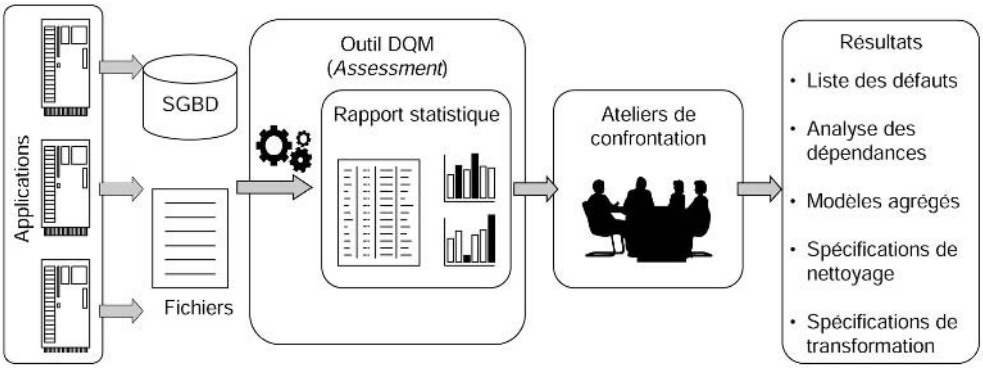

Figure 5.3 – Amélioration de la qualité des données dans un projet de migration de données

L'amélioration de la qualité de la donnée est un processus répété qui doit être mesurable, limité dans le temps (charge contre gains) et validé par les métiers.

Les étapes de la migration s'appuient donc sur des outils de DQM issus de l'offre des grands éditeurs ETL (Informatica, IBM, SAS, SAP/BO…). On combine en fait les outils d'ETL (Power Center pour Informatica, Datastage pour IBM), de *profiling* (Data Explorer pour Informatica, Information Analyser pour IBM) et de qualité des données (Data Quality pour Informatica, Qualitystage pour IBM). Le principe est, dans un premier temps, de transférer les données à migrer dans une base intermédiaire et de les analyser. Cette analyse demande un travail préparatoire de chargement et de normalisation des données. On génère ensuite un rapport statistique dont chaque écart pourra être challengé en atelier afin d'établir des actions correctrices.

La deuxième étape consiste à améliorer la qualité de ces données *via* les mécanismes évoqués de normalisation, déduplication, consolidation et enrichissement. Ces données de meilleure qualité sont ensuite stockées dans une nouvelle base intermédiaire qui peut être utilisée lors de la migration.

Bien entendu, ce processus est à renouveler, les règles et dictionnaires (voir ci-après cette notion) étant progressivement enrichis à l'aide des métiers. Ces derniers sont aussi sollicités pour arbitrer dans tous les cas qui ne peuvent être automatisés.

La dernière étape est le chargement de la base de données de l'application cible. En ce qui concerne le MDM, il est bon d'anticiper une telle démarche, au moins pour le chargement initial.

5.3.3 Zoom sur l'amélioration de la qualité des données

Cette phase du processus doit viser quatre objectifs :

- la standardisation (noms, raisons sociales, adresses par exemple) ;
- la détermination des règles de rapprochement et d'identification de doublons ;
- la fixation des règles de fusion ;
- les règles d'exactitude et de complétude.

Standardisation

La standardisation permet de normaliser des données. Elle s'appuie pour cela sur des **dictionnaires** *ad hoc*, comme : type de voie (Rue, Avenue, Boulevard…), civilité (M., Mme, Melle…), statut juridique des entreprises (SA, SARL, SAS, GMBH…). Il est indispensable de mener à bien cette opération avant l'étape de rapprochement.

Rapprochement (scoring)

Il s'agit d'identifier les objets susceptibles d'exister sous plusieurs instances ou enregistrements dans un même fichier ou une même base de données en vue de leur dédoublonnage : ce sont soit des objets identiques, soit des objets équivalents. Cette tâche se fait par hypothèses et vérifications successives à partir de critères de rapprochement fournis par le métier.

On utilisera soit des règles simples, basées sur des clefs identifiées ou des attributs précis (règles déterministes), soit des règles plus complexes s'appuyant sur plusieurs attributs (règles probabilistes). Cette approche déterministe repose sur une comparaison réalisée sur de gros volumes de données et sur les analyses d'écarts statistiques entre les valeurs ou les chaînes comparées avec l'ensemble des données détenues.

Les comparaisons sont renforcées par l'usage d'algorithmes phonétiques permettant la lecture des chaînes de caractère non plus lettre à lettre mais phonème par phonème. Ainsi, une erreur de saisie peut être gommée ou des noms proches identifiés (par exemple, Dupont et Dupond).

Cette démarche produit des scores statistiques. Le score obtenu peut déclencher, en fonction du dépassement de seuils prédéterminés, des actions de fusion automatique, des processus humains, le rejet de doublons. La définition de ce qu'est un doublon est établie par le métier.

Par exemple, dans un système de facturation et de recouvrement, deux « payeurs » sont considérés comme un doublon si leur adresse et leurs coordonnées bancaires sont respectivement identiques. En l'occurrence, le nom du payeur n'est pas retenu comme critère de rapprochement.

Définition des paramètres de fusion (matching)

On cherche les meilleurs paramètres sur un échantillon connu dans lequel les doublons sont déjà identifiés, on utilise ensuite ces paramètres sur l'ensemble des données :

- création d'indicateur sur les doublons (par exemple, fréquence par type) ;
- pour chaque type de doublon identifié, on spécifie les règles de fusion : les attributs conservés, les seuils de déclenchement automatique de la fusion ou d'alerte pour un traitement manuel, les sources préférentielles.

Exactitude et complétude

Les règles effectives sont à définir au cas par cas. Elles sont principalement utilisées en décisionnel. En MDM, exactitude et complétude sont vérifiées par l'application référentielle.

5.3.4 Positionnement DQM versus MDM

MDM et DQM correspondent à des solutions bien séparées et complémentaires.

Le DQM permet le nettoyage des données (normalisation, consolidation, enrichissement, surveillance, analyse, profilage) conformément à des règles spécifiques de gestion de la qualité. Le périmètre des données traitées est plus large que celui des données de référence (données client, financières, produits) : il peut s'agir de tout type de données provenant de sources hétérogènes.

Le **MDM** représente les modèles communs des données de référence, transverses au niveau métier et organisation. Il met à disposition des applications une vue fiable,

exhaustive et à jour de la donnée de référence. Le MDM, en tant que source de vérité unique des données, nécessite un processus de qualification des données avant intégration et une mise à disposition des processus et des applications du SI**.**

En conclusion, le **DQM est une brique complémentaire, souvent essentielle au MDM** pour fiabiliser l'information, et il intervient en ce sens dans les différents stades d'acquisition et d'enrichissement de la donnée de référence : **assainissement des données lors de l'acquisition (normalisation et consolidation des données créées ou modifiées, déduplication), validation de la conformité tout au long du cycle de vie de la donnée de référence.** Le DQM permet donc au noyau du référentiel de disposer d'une donnée de référence fiable. Cette complémentarité est telle que les outils MDM prennent pleinement en compte la dimension DQM.

Ainsi, le marché du MDM se divise entre les outils qui fonctionnent au travers de connecteurs (EBX Platform d'Orchestra Networks), ceux qui offrent des connecteurs et intègrent la démarche qualité dans leur processus de gestion (MDM Server d'IBM), ceux qui intègrent certains algorithmes de normalisation et de rapprochement ainsi que des connecteurs d'accès à des fonctions plus sophistiquées (SAP MDM), ceux enfin qui offrent des fonctions intégrées complètes (Initiate Systems). On remarquera que les outils spécialisés dans la gestion des tiers (personnes physiques, personnes morales) sont, en moyenne, mieux dotés.

La figure 5.4 schématise cette complémentarité et décrit le processus de traitement d'un objet entre le MDM et le DQM.

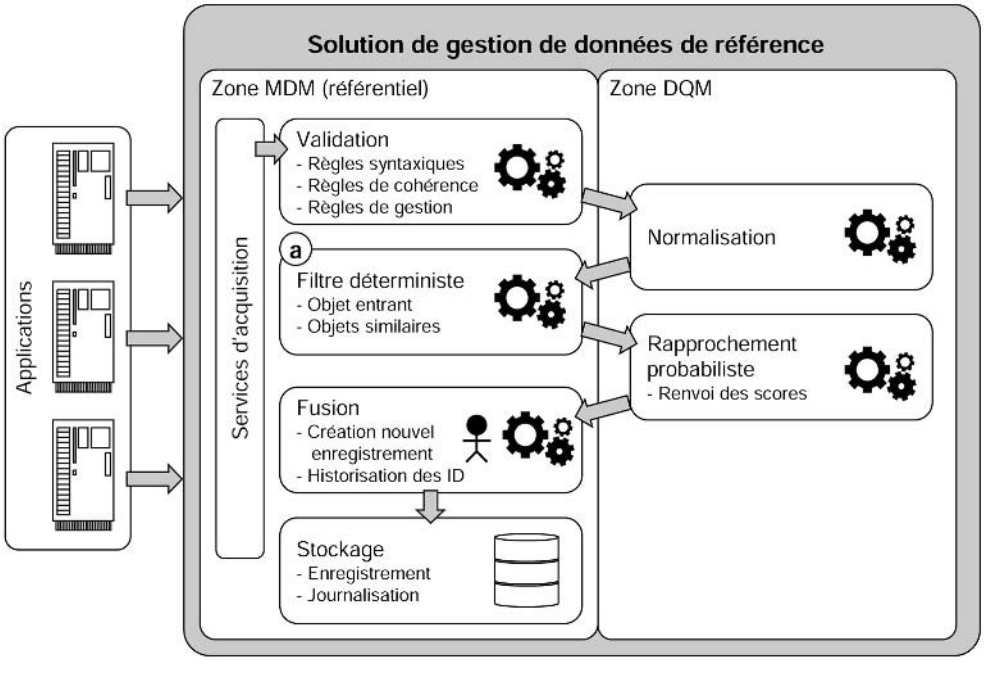

Figure 5.4 – Complémentarité DQM et MDM

On observe dans la figure 5.4 que le filtre déterministe est une étape qui dépend de l'application utilisée. Certaines applications fonctionnent directement sur des algorithmes DQM probabilistes intégrés à la solution. Les solutions composées d'un outil MDM associé à un outil DQM fonctionnent en utilisant les deux filtres (figure 5.4a).

De nombreuses solutions de DQM sont disponibles, d'Informatica Data Quality à IBM Information Server (Analyser et QualityStage), en passant par Oracle DQM, SAP Business Object Data Quality, Harte-Hanks Trillium Software Data Quality…

5.3.5 Pourquoi le DQM ne peut-il remplacer le MDM ?

Certes une solution DQM permet de disposer d'une information fiable, normalisée et sans doublons. Mais la persistance (stockage), l'historisation, la gestion des identifiants, le cycle de vie et la gestion de la propriété des données qu'elle traite sont hors de son périmètre fonctionnel. Il en résulte qu'une solution de gestion de la qualité constitue un processus de traitement de la donnée avant stockage dans le référentiel (voir figure 5.4).

Un référentiel MDM offre aux processus et applications un point unique de vérité pour la donnée de référence.

À l'inverse, il est possible de créer un référentiel qui centralise les données de plusieurs sources d'acquisition, sans pour autant traiter la qualité des données qui s'y trouvent. Ce référentiel reste une source unique de vérité mais, dans un tel cas, les principes MDM d'information fiable et complète ne sont pas respectés.

5.4 EII (*ENTREPRISE INFORMATION INTEGRATION*)

L'EII a pour objectif de simplifier l'implémentation d'applications utilisant des sources de données disparates. L'EII présente une vue virtuelle de données provenant de sources multiples. Celle-ci peut correspondre aux besoins des applications cibles de l'EII et non plus aux modèles physiques des sources.

En théorie, les applications cibles peuvent accéder aux vues présentées en lecture et écriture. En pratique, les besoins constatés se limitent surtout à la lecture et le mode « mise à jour » entraîne une complexité souvent rédhibitoire pour être couramment implémenté.

Une solution EII doit offrir :

- aux sources :
  - des connecteurs, proposés par l'éditeur ou spécifiques, adaptés à chaque source de données. Les connecteurs fonctionnent en mode *pull* (l'EII demande la donnée), sans gestion d'événements ;

– des fonctions de *mapping* (mise en correspondance de modèles) et de transformation des données. Une interface graphique est souvent proposée pour construire les mappings et définir les règles de transformation ;

- des technologies pour l'accès aux données : XQuery, JDBC, ODBC, Web Services sont les plus utilisées pour requêter l'EII.

La figure 5.5 résume les fonctionnalités de l'EII.

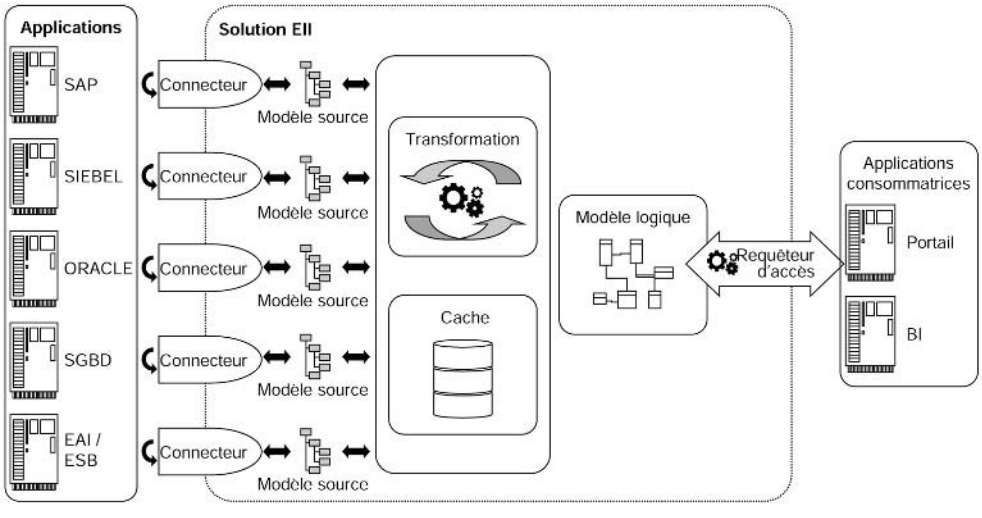

Figure 5.5 – Fonctionnalités EII

5.4.1 Positionnement EII versus MDM

MDM et EII correspondent à des solutions bien séparées et complémentaires.

Le MDM a pour objectif l'harmonisation et la distribution d'une donnée de référence correspondant à la seule vérité admise dans l'entreprise.

L'EII restitue les données à l'identique de leur source sans harmonisation, ni réconciliation, ni correspondance à une vérité unique. L'EII est un *hub* de données ayant pour vocation de servir de *pool* de données pour un ensemble déterminé d'applications. Pour construire le modèle virtuel proposé aux applications cibles, l'EII peut utiliser les données de plusieurs référentiels sources ainsi que des données transactionnelles provenant d'autres sources.

5.4.2 Pourquoi un outil EII ne peut-il servir de référentiel ?

Le principal objet du MDM est l'harmonisation et la qualité des données répondant à un modèle unifié et partagé par l'ensemble des applications utilisant les données en question. Au contraire, l'EII exclut les traitements qualité dans la transformation des

données. Ainsi dans l'EII, le modèle logique présenté aux applications cibles correspond à leurs besoins et non aux besoins d'une vision unique et partagée. De plus, l'EII est une version technique de l'accès aux données et n'offre aucune capacité pour la gouvernance.

Profitons enfin pour éliminer les autres outils techniques d'intermédiation dans le contexte de gestion des données de référence**. Les EAI, ETL et ESB servent au transfert de données, et non à leur gestion**. Nous reviendrons cependant sur leur importance plus tard.

5.5 ANNUAIRES

Les annuaires permettent de gérer des données de référence de type personnes et organisations, en général dans l'optique :

- d'authentifier des utilisateurs ;
- de définir leurs droits ;
- de constituer un carnet de contacts (téléphone, mail, fax, adresse…) ;
- de lier les utilisateurs et leurs ressources techniques (matériel, poste de travail…) ;
- de lier les utilisateurs et leur entité organisationnelle (site, organisation juridique, organisation commerciale…).

Le standard est LDAP (*Lihgtweight Directory Access Protocol*).

Le modèle de données LDAP est de type hiérarchique, les données sont structurées en arbre. Le langage de manipulation des données impose de connaître la position d'un objet dans cet arbre. Il n'y a donc pas d'indépendance entre les programmes qui manipulent les données et ces données, ce qui représente un inconvénient majeur : toute modification dans la structure de l'arbre doit être répertoriée dans les programmes. Il faut donc choisir une structure des données très stable dans le temps, bien indépendante en particulier des organisations.

5.5.1 Positionnement annuaire versus MDM

Un annuaire LDAP répond donc à un protocole permettant l'accès à l'information. Il est plus efficace pour la consultation, ses accès en lecture sont beaucoup plus performants qu'en écriture. Il est doté de mécanismes facilitant la recherche d'informations et la présentation structurée des résultats. Le protocole est suffisamment léger pour permettre de grandes performances en consultation.

Le besoin de consultation ainsi que les performances attendues liées à l'utilisation du LDAP nécessitent la répartition des annuaires au sein de l'entreprise. Ils se sont donc dotés de capacités de duplication pour la répartition et la diffusion vers des annuaires locaux.

À l'inverse, des solutions de méta-annuaires permettent la ré-agrégation des données réparties dans les annuaires locaux. Ils peuvent fonctionner en *repository* ou en *registry* (avec copie vers l'instance centrale ou simple indexation pour constitution d'un annuaire virtuel).

5.5.2 Pourquoi un annuaire ne peut-il servir de référentiel ?

Les annuaires ne peuvent pas être considérés comme des référentiels car ils sont plutôt destinés à un usage technique, même si les cas d'utilisation rencontrés dans les entreprises dépassent ce simple cadre. Historiquement, les annuaires ont répondu à des besoins MDM en l'absence de solutions réellement adaptées. Même si les annuaires LDAP répondent en partie à une stratégie MDM (normalisation, définition sémantique) ils ne couvrent cependant pas les processus de gestion et n'offrent aucune capacité de pilotage des données.

Les méta-annuaires n'apportent rien au débat et sont tout autant limités en capacité de gestion.

La gestion et la collecte d'informations sur les tiers, jusqu'ici traitées au travers d'annuaire, doivent être dévolues au MDM. Les annuaires doivent alors être considérés comme des auxiliaires techniques, essentiellement pour l'authentification et la gestion des moyens de communication (mail, téléphone). En effet, les annuaires restent souvent plus performants dans les tâches d'authentification (temps de réponse excellent).

5.6 CRM (*CUSTOMER RELATIONSHIP MANAGEMENT*)

CRM, acronyme de *Customer Relationship Management*, est une application d'entreprise visant à gérer et valoriser les relations entre l'entreprise et ses clients. Le CRM vise non seulement à renforcer la relation immédiate du client à l'entreprise (transaction) mais aussi à établir dans le temps une relation forte entre l'entreprise et chacun de ses clients (rétention). Ceci se décline en plusieurs spécialités, telles que le marketing stratégique ou opérationnel, mais aussi les services après vente ou la gestion des forces de vente.[1] Le CRM vise donc l'amélioration de processus tels que :

- la gestion des contacts entrants et sortants (propositions, demandes, réclamations…) ;
- le suivi des prospects ;
- la maîtrise de la connaissance client ;
- l'analyse et la prévision des ventes ;
- l'analyse des campagnes marketing/communication ;
- le télémarketing, etc.

1. R. Lefébure et G. Venturi, *Gestion de la relation client*, Eyrolles, 2005.

Il existe différents progiciels ou modules d'une suite de CRM en fonction des processus ciblés.

5.6.1 Positionnement CRM versus MDM

Le CRM est utilisé pour opérer des processus métier en relation avec le client, non pour gérer la donnée client. Généralement, le CRM est utilisé par des profils marketing ou vente. Les informations entrées correspondent donc à un besoin opérationnel, avec des contraintes d'utilisation inhérentes à ces métiers.

Les grandes entreprises mettent le plus souvent en œuvre un CRM par ligne métier, en fonction des spécificités de chacune d'entre elles. La diversité des types de canaux de liaison avec le client est une seconde source de multiplication des outils associés au CRM (particulièrement dans les entreprises de services multicanaux comme la banque ou l'assurance).

À notre connaissance, aucune entreprise ne limite la gestion de ses clients aux seules informations marketing et commerciales. La prise en compte d'autres dimensions, au sein d'autres processus ou applications (logistique, comptabilité) est également importante.

Multi-instanciation et limitation à ses propres processus empêchent le CRM de tenir une de ses principales promesses : la connaissance du client (soit une vision à 360 ° du client). Au mieux, on peut utiliser le CRM comme un référentiel de zone spécifique à un métier ou un service.

L'usage opérationnel du CRM à l'échelle d'une grande entreprise est tel qu'il induit, à très court terme, une dégradation de la qualité des informations client détenues (doublons, transgressions fonctionnelles pour la gestion opérationnelle de cas particuliers...).

En revanche, le MDM permet de gérer un unique modèle « tiers » couvrant par exemple les clients, les partenaires et les fournisseurs au sein d'un même référentiel. Le MDM offre aussi une capacité d'évolution du modèle et du support des états de la donnée (en rapport avec le cycle de vie métier), fonctionnalités qui sont aux limites des progiciels de CRM (modèle peu extensible, états alignés sur les processus outillés).

5.6.2 Limites d'un système CRM

Les outils CRM ne permettent pas de « piloter » les référentiels :

- la gestion événementielle est propre aux processus, pas à la donnée ;
- l'interfaçage est moins adaptable qu'avec le MDM ;
- les règles portées par le CRM contraignent son utilisation en tant que référentiel (imbrication des règles, modèles et processus) ;
- le CRM n'outille pas le *key mapping* (cohérence de clefs de transcodification, historisation des clefs dédoublonnées) ;

- la vision à 360 ° demande l'agrégation d'informations de plusieurs sources (multiples CRM, autres systèmes) ;
- l'évolution du modèle sera alourdie car contrainte par l'outil ;
- le passage en production d'un besoin devra se faire au rythme des mises à jour des développements CRM, c'est-à-dire quelques fois dans l'année ;
- la mesure de la qualité et l'établissement de tableaux de bord propres à la donnée sont mal aisés…

Le CRM se propose notamment de lutter contre l'attrition des clients (perte de ces derniers) grâce à une meilleure connaissance de ceux-ci. Aujourd'hui, le CRM a montré ses limites dans la poursuite de cet objectif. Il peut conserver son rôle opérationnel, c'est-à-dire la maîtrise des processus, tandis qu'il est nécessaire de mettre en place des solutions référentielles spécifiques (CDI : *Customer Data Integration*) pour renforcer la connaissance des clients par la consolidation des informations les concernant.

5.7 PLM (*PRODUCT LIFECYCLE MANAGEMENT*)

Le PLM (ou gestion du cycle de vie produit) est une application d'entreprise qui vise à créer, gérer et partager l'ensemble des informations de définition, de fabrication et de maintenance d'un produit industriel, tout au long de son cycle de vie, depuis les études préliminaires jusqu'à la fin de sa vie.

Le PLM est une extension du PDM (*Product Data Management*, dénommé en français GDT pour *gestion des données techniques*). Ce dernier est une solution de gestion des informations techniques associées au produit lors de la phase de conception (fichiers CAO, plans, documentation...). Il s'agit du coffre-fort technique d'une entreprise.

Le PLM couvre en fait un domaine plus vaste et s'organise autour d'un système d'information comprenant la CAO (conception assistée par ordinateur), la GDT (gestion de données techniques), la simulation numérique, la FAO (fabrication assistée par ordinateur), le KM (*Knowledge Management*). De ce fait, il permet également la gestion des processus de l'entreprise (*workflows*) en relation avec cette documentation. Parfois intégré avec un ERP, il assure des échanges d'informations en temps réel entre le département de conception et les départements de gestion et de fabrication.

Le PLM gère deux aspects importants et complémentaires du cycle de développement d'un produit : les organisations, les processus et les méthodologies d'une part, les outils et systèmes d'information mis en œuvre d'autre part.

Les principes mis en œuvre sont au nombre de cinq :

- l'approche du marché orienté processus pour optimiser les procédés dans les secteurs d'activités spécifiques ;

- les espaces de travail collaboratifs pour intégrer la communication et la collaboration via un environnement 3D commun ;
- le modèle PPR (Produit, Procédé, Ressources) pour relier les représentations du produit ;
- les ressources de fabrication (outillage, usine, opérateurs) et les procédés de production ;
- la connaissance pour conserver, partager et réutiliser les données de l'entreprise, le savoir-faire et le capital intellectuel.

5.7.1 Positionnements PLM et MDM

MDM et PLM couvrent des périmètres de données un peu différents et sont largement complémentaires. *A priori*, seules les activités du secteur industriel sont concernées par la question du choix entre PLM et MDM et, en particulier, les éventuels recouvrements des données au niveau de la production (manufacturing) (figure 5.6).[1]

Le PLM gère toutes les informations dédiées aux produits dans le détail, données et processus afférants, transverses aux niveaux métier et organisation. Ainsi le PLM est censé tout faire, tout stocker et tout gérer.

Le MDM se focalise pour sa part sur les données de référence, **transverses au niveau métier et organisation**. C'est un outil de gestion et de gouvernance.

Dans le cas où le PLM est déjà implémenté, l'intérêt du MDM est réel :

- si le PLM est multi-instancié et si des conflits existent entre ces instances ;
- lorsque des données produits sont aussi créées dans d'autres systèmes appartenant à d'autres métiers non esclaves du PLM (logistique pour la *supply chain*, mais aussi finance, comptabilité et enfin marketing) ;
- lorsqu'il existe un besoin d'harmonisation avec les informations périphériques aux produits : fournisseurs, prix, catégories de *reporting*…

La figure 5.6 schématise le positionnement du MDM par rapport au PLM.

1. Sur la comparaison entre MDM et PLM voir les travaux de Tech-Clarity, maintenant racheté par le groupe Aberdeen : http://www.aberdeen.com/
Voir notamment les travaux de Jim Brown, dans le document « The Role of Pim and PLM in the Production Information Supply Chain : Where is your Link ». http://www.tech-clarity.com/documents/Roles_of_PIM_and_PLM.pdf

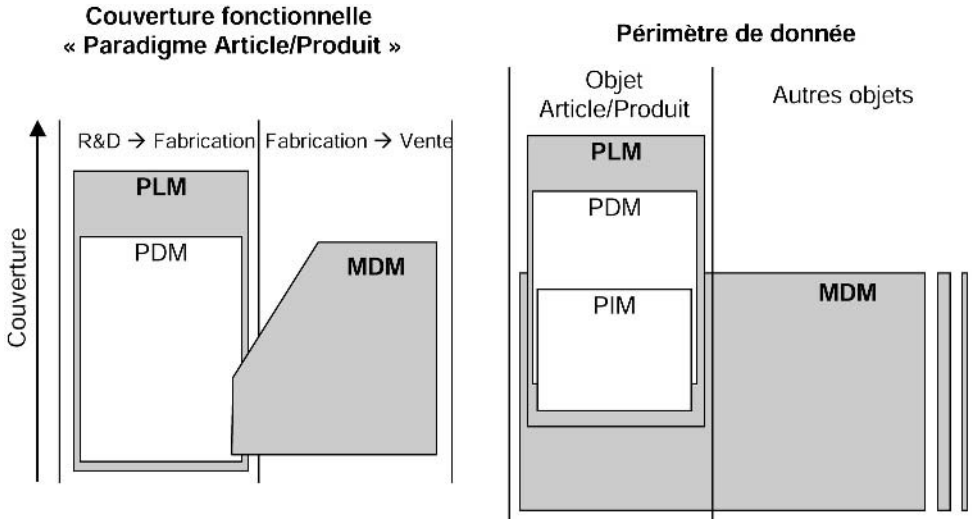

Figure 5.6 – Positionnements PLM et MDM

On veillera donc particulièrement à l'interconnexion du MDM et du PLM au niveau des activités de production et en particulier concernant la chaîne d'approvisionnement.

5.7.2 Les fonctions du PLM

Le périmètre du PLM, déjà esquissé, est très large. Il inclut une approche stratégique du développement de produit et de ses processus, les outils mis en œuvre dans la création ou la diffusion de données, le travail collaboratif…

En fait, la définition est si large qu'elle peut s'appliquer à de nombreux points de vue et que beaucoup de démarches ou d'outils différents peuvent s'y référer, ce qui produit parfois une certaine confusion dans l'identification de ce qui relève ou non du PLM. On comprend alors que l'on peut rencontrer autant de définitions du PLM et de son périmètre qu'il y a d'acteurs dans le monde du développement de produit, de ses processus, de la gestion de projet, du travail collaboratif, de la gestion des données....

Une démarche PLM requiert une vision à long terme des axes de développement de l'entreprise dans ses activités de production. Les systèmes d'informations et outils qui permettent sa mise en œuvre doivent gérer une complexité telle que ce périmètre et cette couverture fonctionnelle sont en fait rarement atteints. De nombreux problèmes sont encore à résoudre, de nombreux modèles restent à définir…

La démarche pourra alors se recentrer sur les besoins spécifiques d'une branche ou d'un métier. On peut parler alors de PLM orientés « processus ».

Ce que fait un système PLM

Résumons notre propos en détaillant les fonctions d'un système PLM :

- fédérer et intégrer plusieurs aspects du développement d'un produit et de ses processus ;
- gérer la documentation technique par l'intermédiaire d'un ou plusieurs coffres-forts centralisés auxquels tous les services peuvent accéder : BE, BM, marketing, maintenance...
- gérer les processus clés : demande de modification (*engineering change request*), réalisation d'une modification (*engineering change order*)...
- gérer la composition et la structure d'une gamme de produit et de toutes ses variantes.

Ce que ne fait pas un système PLM

- gérer la production ou les ordres de fabrication ;
- gérer la relation client ;
- gérer les ressources humaines.

5.8 SYNTHÈSE

Le tableau 5.1 synthétise les différentes solutions évoquées dans ce chapitre. Rappelons que certaines solutions restent spécifiques à tel ou tel type de donnée (article ou produit pour le PLM et tiers pour les annuaires et CRM).

Tableau 5.1 - Positionnement des différentes solutions (MDM, EII, PLM, DQM)

Fonction recherchée	MDM	DQM	EII	Ann	PLM	CRM	Commentaire
Harmonisation de la donnée	***	-	*	**	**	*	MDM : point différenciateur des autres solutions et particulièrement avec le PLM, le CRM et les annuaires : tout type de données de référence.
Description des données et processus	***	**	-	**	***	**	PLM : intégration d'outil de gestion de processus pour le cycle de vie des produits uniquement. CRM : bonne couverture des processus, limitation au niveau des données. Annuaire : bonne description des données, pas de processus. DQM : capacité à centraliser les règles métier pour orchestrer les processus qualité des flux de données. MDM : capacité de modélisation et d'exécution des processus d'administration des données de références et non des processus métier. Attention : certaines solutions ne sont pas orientées collaboratif donc pas de processus.
Qualité de la donnée	**	***	-	-	*	*	DQM : capacité d'adresser l'ensemble des problématiques de qualité des données du SI (données clients mais aussi financières ou produit). MDM : capacité d'adresser le maintien de la qualité des données au périmètre de la solution.
Pilotage & tableau de bord	**	***	*	-	*	**	DQM : capacité de fournir des indicateurs de pilotage sur la modification des données. MDM : Idem DQM. CRM : pilotage orienté processus de gestion client mais pas donnée.

Fonction recherchée	MDM	DQM	EII	Ann	PLM	CRM	Commentaire
Gestion des données techniques	N/A	N/A	N/A	N/A	***	N/A	PLM : le facteur différentiateur avec les autres solutions.
Gestion de versions/gestion de configuration	**	N/A	N/A	N/A	***	N/A	PLM : capacité à gérer l'historique des modifications et à faire la gestion de configuration. MDM : capacité à gérer l'historique des modifications de certains packages.
Virtualisation	N/A	N/A	***	**	N/A	N/A	EII : les solutions proposent toutes des accès aux données suivant les standards Objet, XML et relationnels. Point fort de la solution. Annuaire : possibilité des méta-annuaires mais limités aux données des tiers.

5.9 COMPLÉMENT D'INFORMATION SUR L'EIM

L'**EIM (*Entreprise Information Management*)** recouvre l'ensemble des pratiques et outils permettant la gestion et la mise à disposition pertinente et complète des informations détenues par l'entreprise. Dans notre acception, elle englobe donc le MDM. L'EIM impose une réflexion plus orientée sur le contenu que les contenants (techniques, formats ou normes). **Du point de vue des outils, l'EIM** constitue plus un *framework* (cadre applicatif) qu'une solution. Plusieurs outils, technologies et techniques, y concourent et garantissent l'inter-opérabilité des systèmes. L'EIM recouvre des champs tels que :

- le MDM (incluant la qualité de la donnée) ;
- les solutions d'indexation et de recherche d'information ;
- les outils de collaboration ;
- les solutions de gestion documentaire ;
- les solutions techniques (annuaires, répertoires de métadonnées, outils d'intermédiation, SOA…) ;
- et par extension l'ensemble des outils de gestion des données ou des informations (gestion électronique des documents, gestion des connaissances).

Si l'EIM ne constitue pas le propos principal de cet ouvrage, nous encourageons cependant à garder cette notion à l'esprit car MDM et DQM sont une constituante essentielle du passage de la gouvernance des données à la gouvernance des informations.

En résumé

La gestion des données de référence peut être dévolue à divers types d'application (MDM, DQM, EII, Annuaires, PLM, CRM). En fonction du type de donnée et du cadre d'utilisation, chacune de ces applications possède ses avantages.

L'utilisation de tel ou tel outil en lieu et place d'un référentiel dédié ne pourra cependant être envisagée qu'en connaissance des limites d'un tel choix.

Pour les initiatives d'importance, notre approche favorise les outils MDM et DQM pour leur transversalité et leurs fonctions de gestion et de pilotage.

6

Architecture fonctionnelle du MDM

Objectif

Ce chapitre aborde les solutions MDM en tant que telles. Il décrit les fonctions attendues au périmètre de telles solutions et propose ainsi une grille d'analyse des solutions disponibles sur le marché.

Nous indiquons aussi les typologies des offres de solutions sur le marché (CDI, PIM…) et présentons la notion de socle référentiel.

6.1 FONCTIONNALITÉS D'UNE SOLUTION DE GESTION DE DONNÉES DE RÉFÉRENCE

Une solution de gestion de données de référence n'est pas seulement une solution technique. **Elle répond à un besoin de gouvernance des données**.

Ainsi, les fonctionnalités attendues ne sont pas simplement des fonctions de collecte, stockage et diffusion mais répondent aussi à un besoin de gestion de la performance en termes de qualité (intrinsèque et services) et sont soumises à une maîtrise claire des responsabilités (gestion des acteurs).

Nous verrons dans la troisième partie de l'ouvrage que l'organisation en charge de la donnée est aussi importante que la solution informatique qui outille cette gestion. La solution doit donc supporter des capacités de suivi, en relation avec les objectifs du cadre de gouvernance.

Ainsi, la mise en œuvre d'une solution de MDM incite l'architecture d'entreprise **à définir de nouvelles règles ou à étendre l'outillage existant afin de garantir le bon usage de la solution MDM**.

La figure 6.1 propose une représentation des couches fonctionnelles d'une solution de gestion de données de référence se déclinant en sept couches.

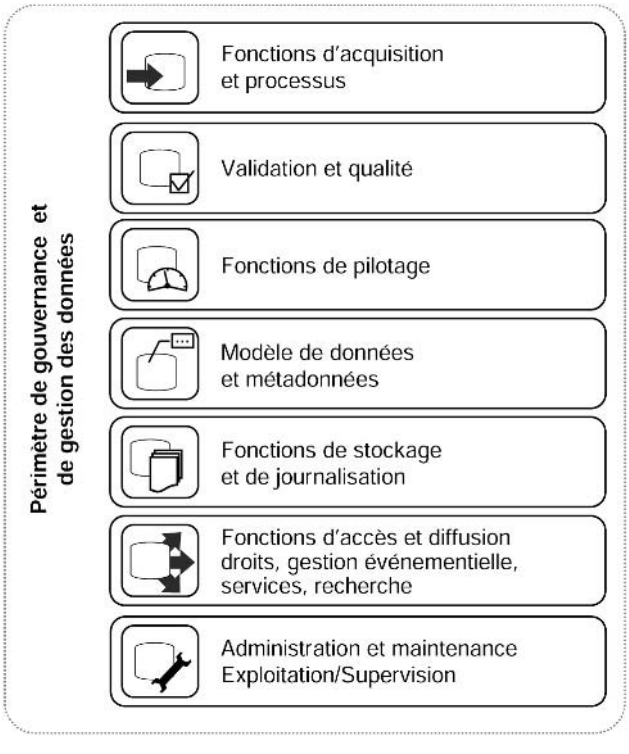

Figure 6.1 — Fonctionnalités d'une solution de gestion de données de référence

Voyons plus en détail ces différentes fonctions.

6.1.1 Acquisition de la donnée et processus

Saisie de la donnée

La solution doit permettre une acquisition directe de la donnée. A *minima*, elle propose des écrans de saisie, qui sont généralement autogénérés en fonction du modèle de donnée. Au mieux, elle intègre des capacités de modélisation et d'exécu-

tion de *workflow* (c'est-à-dire un enchaînement de tâches à réaliser). Cette capacité est d'autant plus importante si la saisie complète de la donnée échoit à plusieurs services ou organisations. Par exemple, un compte comporte des éléments relevant de la comptabilité, de la finance et de la gestion.

Toutes les solutions MDM ne proposent pas de *workflow*, et celles qui en offrent se limitent à des fonctions basiques bien en deçà d'un outil de BPM classique. Pour cette raison, on peut être amené à utiliser un tel outil en réponse à un processus d'acquisition complexe. On appréciera les solutions offrant notamment la fonction de délégation de droits.

Acquisition technique

Une solution référentielle doit permettre l'acquisition de données depuis des sources externes aussi bien sous forme de message unitaire que par chargement en masse.

Les protocoles techniques répondant à ces besoins peuvent être divers mais il reste à la charge de la solution d'assurer la persistance des données après en avoir acquitté la réception. Ceci est vrai quelle que soit la validité de la donnée concernée. Ainsi, la gestion et la conservation des états de la donnée en attente de validation doivent être assurées par la solution (enregistrements des états latents ou *stagging*)

Gestion des relations et hiérarchies

Une des valeurs ajoutées des solutions MDM est de pouvoir rapprocher des informations entre elles. Ces rapprochements sont de deux ordres :

- liens entre objets ;
- rattachement hiérarchique (et groupe).

Les liens (ou relations) entre objets permettent de reconstituer des visions telles que les cellules familiales au sein d'un référentiel client, les liens composés/composants dans un référentiel article ou encore les déclinaisons logistiques des référentiels produits.

Nous préférons les solutions qui offrent des liens entre objets porteurs de sens car la valeur consentie au lien permet la reconstruction d'une vision hiérarchique entre les données. Ainsi, au sein d'une cellule familiale, on peut identifier les grands-parents, les parents et les enfants. Dans un référentiel de personnes morales, on peut reconstituer les filiations groupe, maison mère, filiales pays et établissements de vente.

Les rattachements hiérarchiques et groupes permettent le rattachement d'un objet à une hiérarchie ou nomenclature à N niveaux (s'il y a un seul niveau, on parle alors de groupe).

Pour les hiérarchies ou les nomenclatures, il peut s'agir par exemple du type de rattachement qui s'opère quand on inclut un article dans un catalogue. Le catalogue est une structure hiérarchique normée indépendante de l'article. L'article y est ratta-

ché et les rattachements peuvent être multiples ou uniques en fonction de la liberté qu'on offre à la structure du catalogue. Bien entendu, plusieurs catalogues peuvent être définis sur une même base d'articles, répondant à de multiples navigations ou structurations de l'offre.

Les hiérarchies sont également utiles quand on rattache des contacts clients à une structure d'administration des ventes (par exemple : clients commande, clients vente, clients facturation, clients paiement, clients livraison).

Les groupes reposent quant à eux sur un point de convergence unique et sont utiles pour segmenter le référentiel (hommes, femmes, hyperactifs…).

Tous ces types de liens et de relations sont particulièrement utiles pour trier, segmenter ou encore chercher dans le référentiel.

Prenons l'exemple d'un client point de vente au sein d'un grand groupe de distribution (figure 6.2).

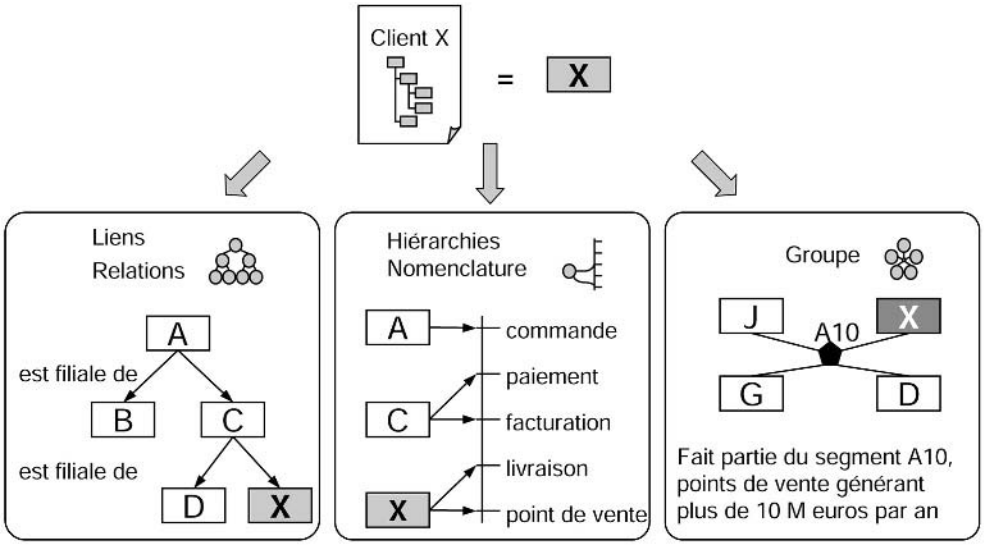

Figure 6.2 – Exemples de liens, hiérarchies et groupes

Les liens et hiérarchies induisent des contraintes techniques spécifiques dans leur mise en œuvre quand on a affaire à de forts volumes. Par exemple, dans l'industrie automobile, prenons le cas des pièces détachées. Si vous lancez une recherche demandant l'ensemble des pièces détachées composant un véhicule, votre recherche devra parcourir des dizaines de milliers de liens avant de constituer une réponse. De même, si pour une pièce définie, vous demandez dans quels véhicules elle peut être utilisée. La modélisation de vos objets ou la création de tables spécifiques d'indexation seront pour beaucoup dans l'amélioration des performances de recherche (voir « recherche et filtre »).

6.1.2 Validation et qualité

La validation

Tout objet entrant dans le périmètre de la solution MDM doit être validé. Cette validation repose sur différentes règles en fonction du cycle de vie métier de la donnée (ainsi, dans le paradigme client, les règles de gestion sont différentes pour un prospect et pour un client actif).

On préférera les solutions offrant une grande souplesse dans la mise en œuvre de ces règles. Il est bon que ces règles soient décrites au niveau de la définition du modèle ou par d'autres règles indépendantes et programmatiques, librement appelées par les services de validation, ou encore par des règles portées par un système tiers (autres référentiels ou moteur de règles).

On citera au moins trois types de règles :

- règles syntaxiques ;
- règles de gestion ;
- règles de cohérence.

Les règles syntaxiques reposent sur la vérification du format de l'attribut (par exemple : date, chaîne alphanumérique de six caractères…). Généralement définies au niveau du modèle, elles sont aussi vérifiées par les outils d'intermédiation.

Les règles de gestion sont portées par un algorithme de vérification de la valeur d'un attribut ou de plusieurs attributs. Par exemple, le volume d'un article ne doit pas être supérieur à 1 m^3 en s'appuyant sur les attributs hauteur, largeur, et profondeur. On inclut au sein du référentiel les seules règles communément répandues. Les applications métier restent seules responsables des règles de gestion métier, on ne les réplique pas au sein du référentiel.

Les règles de cohérence sont décrites par des contraintes entre niveaux de données, c'est-à-dire entre données maître, ou données maître et constitutives ou avec les données de paramètres. Par exemple, on vérifiera le lien entre un objet article et le référentiel fournisseurs, entre l'objet constitutif « profil financier » et son objet maître client ou encore la valeur d'un attribut par rapport à un référentiel de paramètres (devises, pays…). Certaines solutions du marché obligent à supporter toutes les données au sein d'une même instance de serveur pour assurer ce contrôle. Il est préférable d'avoir la possibilité de supporter les différentes données dans des référentiels sur des serveurs distincts.

Traitements qualitatifs

La solution de gestion des données de référence doit permettre le traitement en volume et au fil de l'eau des aspects qualitatifs de la donnée (voir le chapitre sur le DQM).

- **Conformation** – Mise en conformité des formats en fonction d'une norme interne ou externe. Exemple : conformation à la norme ISO/IEC 15434 sur les adresses postales.

- **Dédoublonnage** – Il peut prendre deux formes :
  – déterministe, s'il répond à une reconnaissance de clef(s) ;
  – probabiliste, s'il obéit à des règles complexes, elles-mêmes enrichies éventuellement de dédoublonnages précédents.

L'identification

Générateur d'identifiant unique (centrale d'identification)

Il convient pour l'entreprise d'avoir un identifiant unique afin de permettre une meilleure réconciliation des identifiants sur l'ensemble du SI et plus encore dans le cas d'échanges avec des partenaires (processus transverse entre commercialisateur et distributeur dans le secteur des Utilities, par exemple).

Les solutions référentielles se comportent comme toutes les applications du point de vue des ID (identifiants). Elles génèrent leurs ID propres. Cependant, ces ID ne peuvent pas toujours être utilisés comme identifiants uniques. Deux raisons principales à cela :

- les référentiels peuvent être multi-instanciés. Ainsi, cet ID n'est pas assuré d'être unique ;
- pour les données de référence à durée de vie longue, l'identifiant de référence doit être indépendant de l'application.

Un générateur d'identifiant unique peut ainsi être implémenté au sein de la solution référentielle afin de répondre aux besoins de conciliation. Le référentiel stockera cet ID comme un ID externe, cet ID devient alors le pivot des transcodifications mises à disposition des outils d'intermédiation (voir transcodification dynamique).

La centrale d'identification utilisera un algorithme connu et maîtrisé par l'entreprise, permettant ainsi son portage dans le temps ou la reprise de données en cas de « crash ».

Il est préférable que les identifiants générés par la centrale d'identification soient non signifiants. On transgressera éventuellement cette règle dans le cas de référentiels multi-instanciés afin de donner un indicatif (de zone par exemple) comme discriminant des ID de chaque instance.

L'identifiant d'instance

Une instance (ou occurrence) de donnée évolue dans le temps (modification d'adresse par exemple). Ces évolutions ne modifient pas l'unicité de cette donnée, l'identifiant de la donnée n'est donc pas concerné par une modification sur des attributs non-discriminants. Ainsi, le fait que M. Jean Dupont déménage de la rue des Chaumes à Aix en Provence pour l'avenue du Moulin à huile à Marseille ne remet pas en question son identité.

Nous préconisons d'identifier aussi les instances de donnée à chaque modification. Ainsi, le contenu d'une instance à un instant *t* sera identifié lui aussi de manière unique ce qui permettra notamment son suivi (voir « contrôle de

synchronisation », section 6.1.7), ainsi que l'enregistrement de l'identifiant de l'instance précédente lors de l'historisation. Certaines applications MDM offrent nativement cet identifiant (EBX Platform par exemple).

	Instances / occurrences de données	
	Données historisées	**Données actuelles**
Identifiant d'instance	**735926353782**	**9742637408467**
Identifiant UID	243671890732	243671890732
Civilité	M.	M.
Prénom	Jean	Jean
Nom	Dupont	Dupont
Type voie	**Rue**	**Avenue**
Adresse 1	**Villa des roses**	**Appartement B13**
Adresse 2	**Rue des chaumes**	**Avenue du Moulin à huile**
Code postal	**13100**	**13008**
Localité	**Aix-en-Provence**	**Marseille**

Figure 6.3 – Illustration identifiant d'instance

La transcodification

La mise en œuvre d'une solution de MDM amène à reconsidérer certaines règles acquises du système d'information. En la matière, ces règles concernent généralement l'usage et le périmètre des fonctions auparavant attribuées à d'autres outils de traitement des données (EAI, ETL, *workflow*…). La transcodification est de deux natures :

- transcodification des tables (ou transcodification statique) ;
- transcodification des identifiants (ou transcodification dynamique).

Transcodification des tables (transcodification statique)

Elle assure la transcodification entre N champs d'un objet source et M champs de l'objet cible suivant une grille prédéterminée et finie de correspondances entre les valeurs sources et cibles.

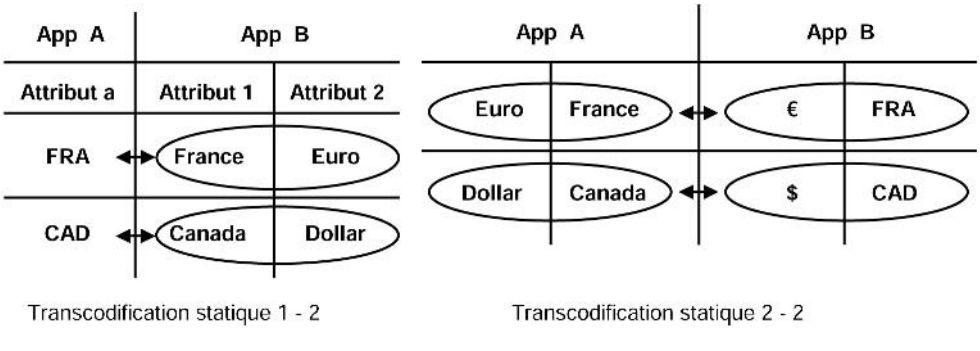

Figure 6.4 – Exemples de transcodifications statiques

Au périmètre d'un projet, la transcodification des tables ne semble pas poser de problèmes architecturaux. Elle semble naturellement supportée et réalisée par les outils d'intermédiation. Au périmètre du SI, la normalisation du *middleware*, notamment dans un contexte multi-référentiels, peut conduire à une analyse différente. Un outil de transcodification est en pratique un référentiel des tables et listes de valeurs de l'ensemble des applications du SI, décrivant les liens entre les champs de tables d'une même catégorie (devises, pays…) . Ainsi, plutôt que de limiter la transcodification statique à l'intermédiation**, il est possible de bâtir un référentiel de tables (ou référentiel de paramètres)** incluant par exemple **:**

- un service de transcodification à destination de l'intermédiation ;
- un service de gestion des tables par IHM ou par fichier d'import/export permettant une gestion centrale de l'ensemble des tables du SI ;
- un service de diffusion des tables à destination des applications du SI ;
- un service de validation des contraintes pour tout processus consommateurs.

Un tel référentiel offrira, de plus, des fonctions standard telles que l'édition des tables (export) ou le pilotage (par exemple, une alerte pour vérification des tables ISO à chaque début de mois).

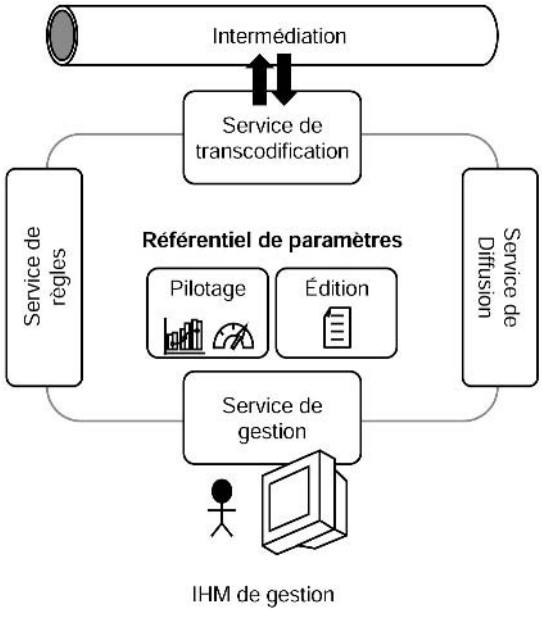

Figure 6.5 – Référentiel de paramètres

En pratique, et par souci de performance, on utilisera le référentiel de paramètres comme outil de gestion et ce référentiel alimentera les outils d'intermédiation, bien plus performants pour la réalisation de l'opération de transcodification.

Transcodification d'identifiants (transcodification dynamique)

Elle porte sur les identifiants d'objets : client ou fournisseur par exemple. Cette transcodification ou ***key mapping*** est la capacité offerte par le référentiel de pouvoir rapprocher ces ID entre eux.

Cette capacité dépend des rapprochements amont et aval offerts par la solution et son architecture de mise en œuvre :

- les rapprochements amonts sont liés aux objets sources et à leurs éventuelles fusions ;
- les rapprochements avals sont liés aux ID des objets applications consommatrices et à leur capture par le référentiel.

Le rapprochement amont repose sur les capacités de « matching » (recoupement) offertes par les algorithmes déterministes ou DQM puis sur le stockage et l'historisation des ID externes (sources) et internes (ID unique).

Pour l'aval, les identifiants ne sont pas prédéfinis et doivent être récupérés au moment de la création d'un identifiant par les applications consommatrices afin de stocker la relation entre les objets échangés.

Ainsi, en fonction du type d'architecture nous aurons les types de rapprochement suivants :

- centralisation – rapprochement aval seulement ;
- coopération – rapprochement amont et aval ;
- consolidation – rapprochement amont principalement, aval si connecté à un système transactionnel.

Le référentiel en tant que source de vérité est maître des identifiants. Cela est notamment sensible avec les référentiels de consolidation ou en coopération. En effet, ces derniers focalisent des données pouvant provenir de sources distinctes, générant potentiellement des doublons. La mise en œuvre d'un outil de DQM et de règles de dédoublonnage permet de fusionner ces objets. Cependant, qu'en est-il des identifiants des objets d'origine ? De plus, les référentiels fusionnent rarement sur un identifiant existant mais déversent les données des identifiants corrélés dans un nouvel objet portant un nouvel identifiant. Enfin, les objets créés dans les applications consommatrices ne peuvent être effacés car ils participent aux transactions au sein de l'application consommatrice.

On préconisera donc de résoudre les rapprochements amont avant d'émettre un objet vers les applications consommatrices. Ceci n'est pas toujours possible, le cas le plus complexe sera celui illustré ci-dessous.

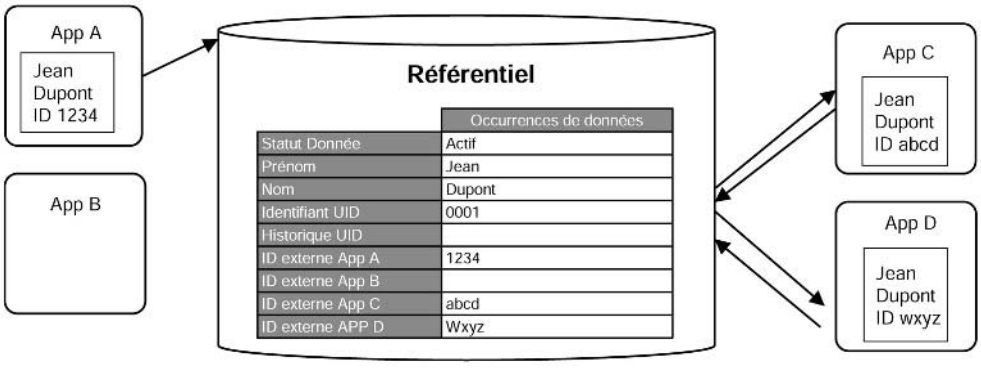

Figure 6.6 – Étape 1 : Simple rapprochement aval sur objet d'origine App A

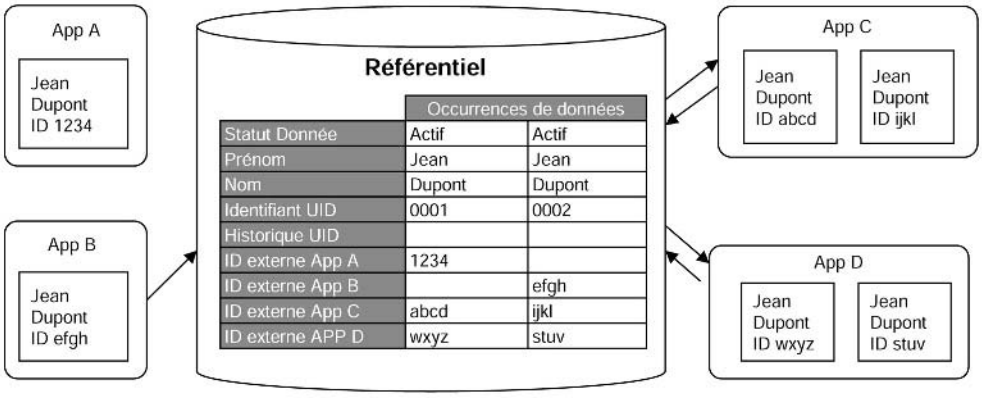

Figure 6.7 – Étape 2 : Simple rapprochement aval sur objet d'origine App B

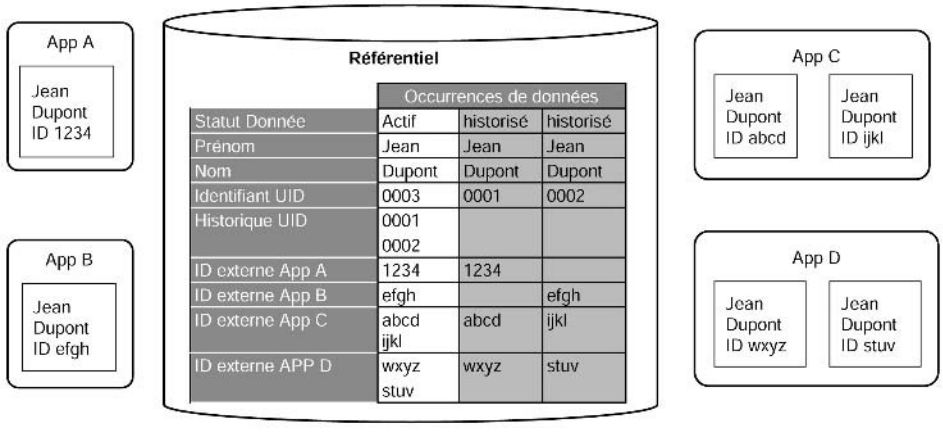

Figure 6.8 – Étape 3 : Rapprochement amont *a posteriori*

De la figure 6.8, on notera qu'il faudra établir une règle de transformation quand une occurrence du référentiel possède plusieurs ID cibles pour une même application consommatrice (diriger vers le dernier ID créé par exemple).

Avant la mise en œuvre de référentiels, l'opération de transcodification était gérée et réalisée par les outils d'intermédiation. L'introduction d'un référentiel remet cette vision en cause. Le référentiel est source de vérité pour les ID donc pour la transcodification. Mais l'opération de *mapping* entre ces ID a lieu au sein de la couche d'intermédiation afin de délivrer le bon ID aux applications consommatrices.

Ainsi, les architectes sont amenés à :

- positionner le stockage et les liens entre identifiants au niveau du référentiel, en offrant un **service de transcodification d'identifiant** aux outils d'intermédiation (**cas A**, attention ce cas peut générer de nombreux appels de services et ainsi générer une contrainte de performance sur le référentiel) ;
- stocker les identifiants dans le référentiel et conserver la réalisation de la transcodification (le *mapping* lui-même) au sein des outils d'intermédiation. Ce qui se traduit par deux sous-possibilités :
  - émettre au sein du message l'ensemble des ID et l'intermédiation fait le tri (solution que nous préférons) (**cas B**) ;
  - ou dupliquer les ID détenus vers la couche d'intermédiation au fur et à mesure de leur consolidation au sein du référentiel (cette dernière solution étant moins conseillée).

Illustrons les cas A et B par les figures 6.9 et 6.10.

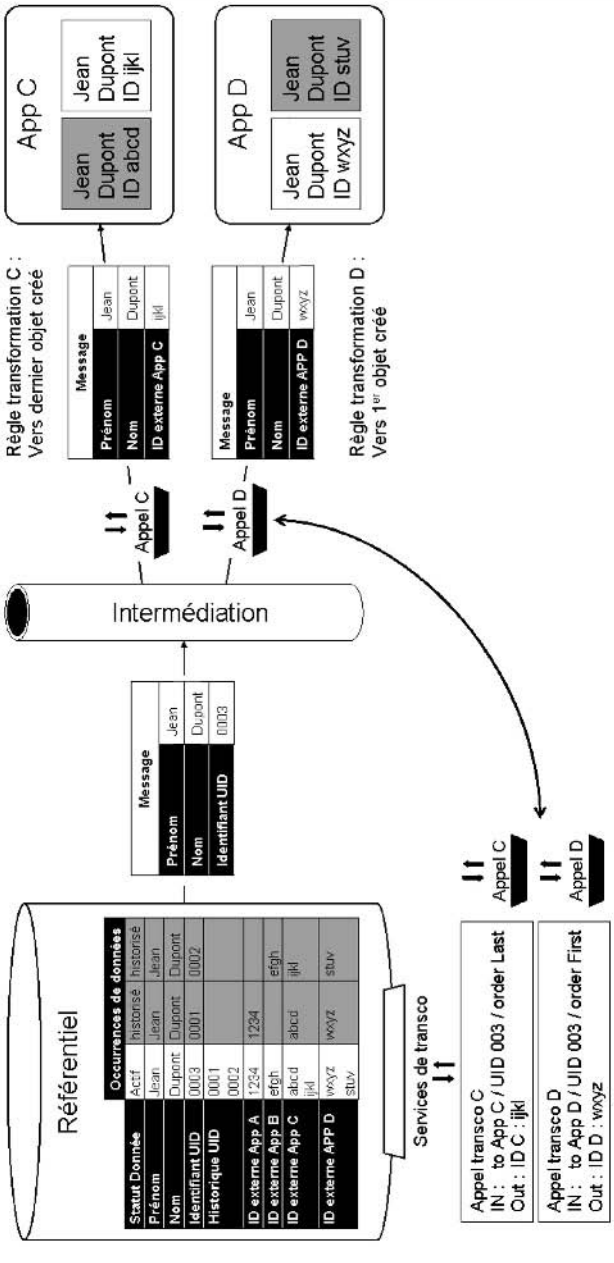

Figure 6.9 – Cas A : service de transcodification sur le référentiel

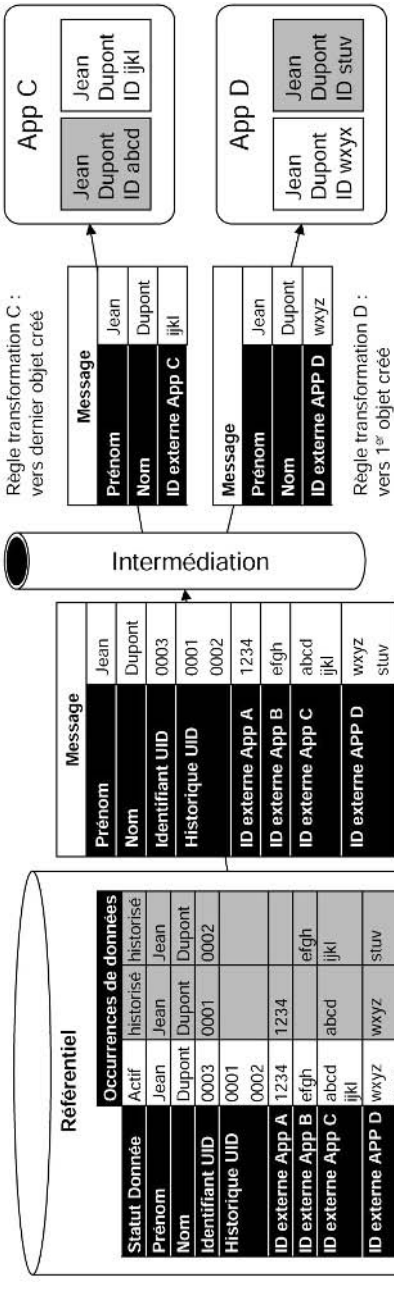

Figure 6.10 – Cas B : émission de l'ensemble des ID et transcodification par l'intermédiation

Hierarchy Mapping

Le *hierarchy mapping* est la capacité de rapprocher entre elles des données ou des familles de données au travers des hiérarchies ou nomenclatures desquelles elles participent ou auxquelles elles sont rattachées.

Par exemple, les catalogues de deux filiales d'un même groupe de distribution peuvent avoir une structure différente mais on veut pouvoir comparer des familles de produits semblables entre elles. On pourra ainsi être amené à rapprocher les structures suivantes afin de comparer les ventes d'eaux en bouteille 1,5 litre :

- filiale A
- Alimentaire/Liquide/Non alcoolisé/Eaux/Non gazeux/1,5 litre ;
- filiale B
- PGC/Liquide/Eaux/Eaux plate/Bouteille plastique/1,5 litre ;
- PGC/Liquide/Eaux/Eaux plate/Bouteille verre/1,5 litre.

Le référentiel doit donc permettre de traiter les informations de l'ensemble des objets rattachés aux structures de catalogue de chaque filiale.

Ainsi, la capacité à gérer de multiples hiérarchies et nomenclatures doit se doubler d'une capacité à les lier entre elles.

6.1.3 Fonctions de pilotage

La mise en place d'une solution de gestion de données de référence répond à des objectifs identifiés au sein d'un cadre de gouvernance des données (voir la troisième partie). Ses effets doivent être mesurables. Cette mesure passe donc par des **indicateurs** et la mise en place de tableaux de bord synthétiques.

Le contrôle opéré par les organes de gouvernance doit aussi pouvoir s'opérer au travers de procédures et de fonctions d'**audit.**

Enfin, la maîtrise des évolutions au périmètre de la donnée repose sur les **analyses d'impact**.

Indicateurs de pilotage et tableaux de suivi

Il s'agit, par exemple, du suivi :

- de la complétude de la donnée ;
- du taux de rejet ;
- des taux de synchronisation (voir aussi administration et maintenance) ;
- d'indicateurs spécifiques métier, etc.

On soulignera deux fonctions de suivi particulières :

- la gestion de l'obsolescence (*decay management*) ;
- la gestion de la confiance (*confident management*).

La gestion de l'obsolescence implique que les attributs surveillés ont un taux d'obsolescence qui augmente avec le temps. Ainsi, une adresse non utilisée ou vérifiée depuis plus d'un an a environ 10 % de chances d'être invalide (les produits MDM de type CDI d'Oracle et IBM offrent cette fonction).

La gestion de la confiance repose sur un indice de confiance attribué à tel attribut (par exemple, en fonction de la source de donnée). En fonction de l'utilisation qu'on souhaite faire de la donnée, on pourra donc s'appuyer sur cet indice pour travailler sur un sous-ensemble d'objets fiables.

Des métadonnées spécifiques permettant l'attribution des pourcentages de confiance devront être paramétrées sous forme de description de l'environnement. On prendra par exemple les groupes d'utilisateurs ou les applications comme critères d'attribution du niveau de confiance et on pourra même croiser ces critères pour établir un indice composite.

Auditabilité

Reposant sur l'historisation et la journalisation (voir section 6.1.5), les capacités d'audit de la solution de gestion de données de référence doivent permettre le contrôle des actions effectuées sur le référentiel. On mettra sous contrôle certaines fonctions et/ou certains attributs. Cet outillage est le support technique des procédures d'audit. Ces procédures peuvent être déclenchées sur alerte ou périodiquement. La composante organisationnelle (procédures) est ici aussi importante que l'outillage.

Là encore, on fera attention à prendre en compte les métadonnées nécessaires à la mise en œuvre de ces fonctions, telles les informations de contexte (date, heure, utilisateurs, système source, UID d'occurrence et ID d'instance…).

Analyse d'impact

L'analyse d'impact repose sur les métadonnées (voir priorité 2 de la section « Gestion des métadonnées », ci-après). En effet, la maîtrise de la description de la chaîne de données doit permettre d'instruire les impacts d'une modification, de processus ou de modèle, aussi bien en amont qu'en aval, que dans le référentiel lui-même.

6.1.4 Modèles de données et métadonnées

Les éléments décrits dans cette section ne seront pas tous outillés par l'application référentielle. Certaines applications complémentaires peuvent être mobilisées pour compléter la solution de gestion des données de référence (*metadata repository*, *business glossary*…), mais les procédures de gouvernance ont aussi une importance capitale.

Gestion des métadonnées

Priorité 1 : assurer la normalisation et la documentation des données de référence

La solution de gestion des données de référence doit permettre la normalisation du modèle de données du paradigme concerné. Suivant les types de solution, l'utilisa-

teur travaille sur un modèle logique (SAP MDM, IBM MDM Server, Initiate sytems…) ou un modèle physique (EBX platform, IBM WCC), la normalisation se fait donc soit sur les modèles physiques, soit sur les contrats d'interface.

Lors de la mise en œuvre, on se concentre d'abord sur les modèles de données : amélioration de leur description fonctionnelle et technique, intégration des règles et contraintes.

Les impacts sur la mise en œuvre concernent :

- **L'organisation** – Le propriétaire (voir cette notion en détail dans la troisième partie du livre) de la donnée peut s'appuyer sur un urbaniste (ou un architecte de données) afin de garantir le formalisme et la cohérence d'ensemble de la donnée.
- **La méthode** – Le processus « Développement » (s'il existe dans l'entreprise) devra renforcer la prise en charge de la description des données au sein des documents de spécifications.
- L'outil qui supporte la modélisation et la conservation des modèles. L'utilisation de cet outil est intégrée au processus « Développement ».

Priorité 2 : décrire puis maintenir la description de la chaîne de donnée

La donnée s'inscrit dans un écosystème. Une donnée de référence peut ainsi s'inscrire dans une chaîne de données potentiellement complexe. Cette notion est bien connue dans les SI décisionnels.

On définit et on conserve chaque modèle et chaque transformation au sein de la chaîne (voir figure 6.11).

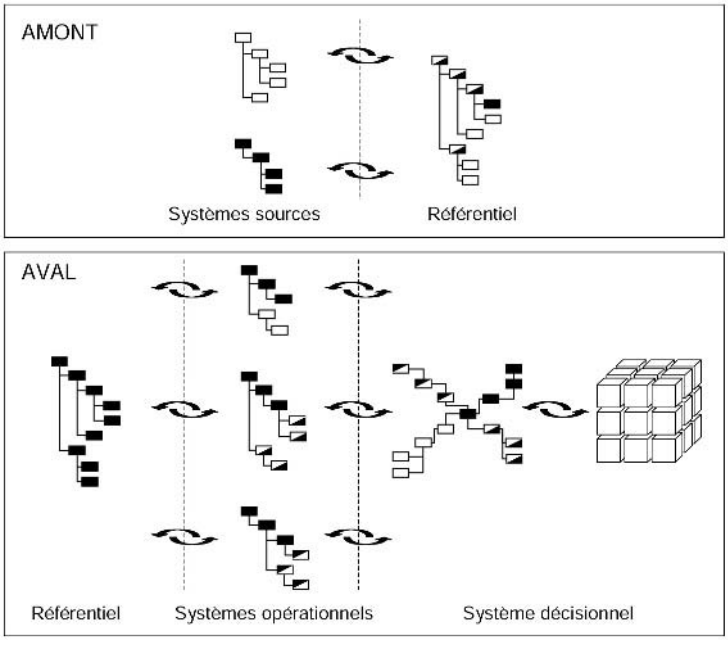

Figure 6.11 – Description des transformations des données dans la chaîne des applications

Les impacts sur la mise en œuvre concernent :

- **L'organisation** – Le propriétaire de la donnée prend en charge les aspects de définition à chacune des étapes puis la maintenance de la chaîne.
- **L'urbanisme** – L'urbaniste (ou l'architecte de données) définit, en soutien du/des propriétaire(s) de la chaîne de données, la cohérence d'ensemble de la chaîne.
  Le même outil support de la modélisation et de la conservation des modèles est utilisé afin d'assurer la traçabilité complète depuis le modèle source.
- **La méthode** – Le processus « Développement » doit établir une étape de renseignement pour les objets et transformations de tous les projets impliqués dans la chaîne**.**

Priorité 3 : définir la sémantique des données et aligner les définitions sur les objets

On incite les métiers à définir leurs concepts métier et à mettre en commun leur vocabulaire spécifique. On aligne les définitions sémantiques sur les descriptions des objets et on veille à cet alignement tout au long de la chaîne de données.

Les impacts sur la mise en œuvre concernent :

- **L'organisation** – Le propriétaire de la donnée prend en charge les aspects d'alignement sémantique.
- **L'urbanisme et le métier** – L'urbaniste (ou l'architecte de données) identifie et priorise les données à définir.
- **La méthode et les outils** – Le processus « Développement » doit établir une étape de garantie de l'alignement sémantique pour les objets référentiels de tous les projets.
- **La conduite du changement.**

La prise en charge de la dimension sémantique impose que chaque métier définisse chaque notion suivant un prisme qui lui est propre. Par exemple, le fournisseur « achat » est un fournisseur titulaire d'un marché, alors que le fournisseur « logistique » est un fournisseur ayant déjà livré une marchandise. La responsabilisation des métiers est nécessaire dans la définition puis l'alignement sémantique au sein de chaque projet.

La mise en cohérence des outils de métadonnées

La connaissance des métadonnées est, en général, dispersée entre divers outils et documents ou disponible auprès d'experts. En effet, chaque étape du cycle de vie d'un projet produit des métadonnées ; chaque intervenant (urbaniste, concepteur EAI, développeur, autres responsables de projet) et chaque projet possède sa propre documentation ou ses propres outils.

Ainsi, la cohérence nécessaire à la maîtrise des principaux objectifs de gestion de la métadonnée peut difficilement être obtenue grâce à l'utilisation d'un outil unique.

Malgré tout, nous préconisons de limiter le nombre d'outils devant supporter l'ensemble des métadonnées (par exemple, le maintien des modèles, de leurs transformations et du dictionnaire sémantique associé). Il convient de renforcer les aspects organisation et méthodologie dans le cas d'une dispersion des outils.

La figure 6.12 illustre la dissémination des métadonnées et le nécessaire alignement des outils et méthodes de gestion des métadonnées.

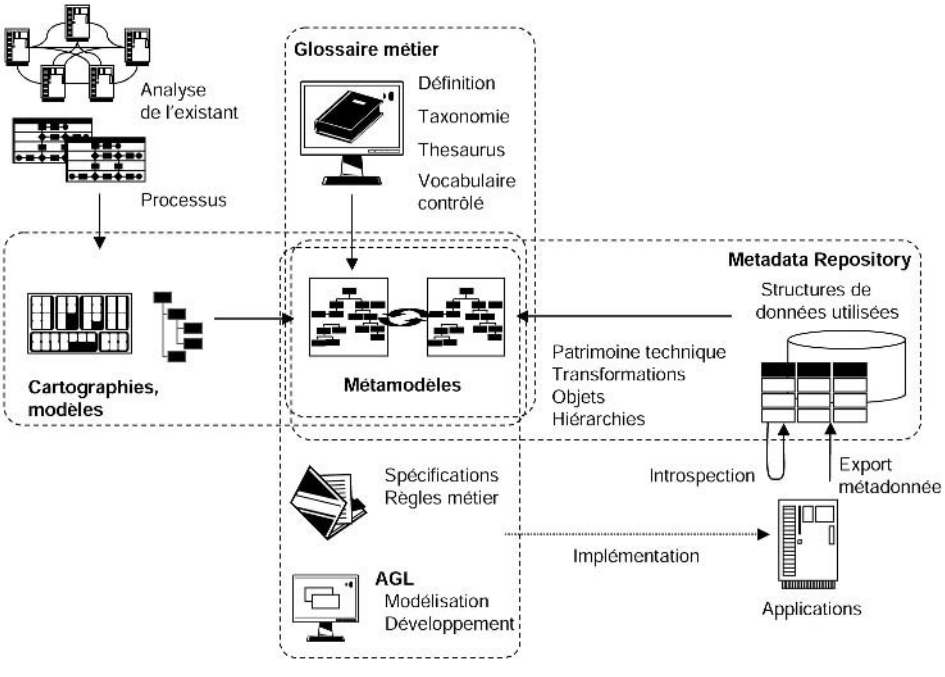

Figure 6.12 – Alignement des outils et méthodes de gestion des métadonnées

6.1.5 Fonctions de stockage et journalisation

Référentiel et contextualisation

La notion de contextualisation (par rapport à un domaine fonctionnel, une organisation fonctionnelle, une organisation géographique…) est un élément indispensable dans la mise en œuvre opérationnelle d'une gestion des données de référence. On peut contextualiser les modèles ou règles et les valeurs.

La contextualisation des modèles et règles, ainsi que la contextualisation des valeurs est affaire de stockage mais elles ont un impact important sur les droits d'accès (voir contextualisation et règles de visibilité).

Certains attributs peuvent être contextualisés, par exemple en fonction d'une zone géographique. Le prix d'un article peut être différent en Europe du Sud ou en Europe du Nord, le nom d'un responsable est optionnel ou obligatoire selon les régions…

Contextualisation des modèles ou règles

- il peut exister un modèle partagé minimum géré dans le référentiel, mais la donnée a un modèle plus complet dans des applications pour chaque domaine fonctionnel (c'est ce que nous avons développé avec la notion de référentiel synthétique) ;
- inversement, il est possible que les applications consommatrices disposent d'un modèle plus réduit. Ainsi la vue offerte pour chacune est plus restreinte que le modèle du référentiel principal ;
- il peut exister des formats ou règles différentes selon les pays ou les organisations.

Contextualisation des valeurs des attributs

Le référentiel doit, bien entendu, offrir des capacités de gestion multilingue (ce qui ne sera pas évoqué ici).

La contextualisation des valeurs s'opère soit par la multiplication des attributs, soit par une fonction de « surcharge » d'un même attribut au sein d'un référentiel décliné en vues, similaires à des sous-référentiels spécifiques.[1] On apprécie aussi la possibilité de choisir si tel ou tel attribut est contextualisable pour une vue particulière du référentiel. Cette fonction ne doit pas générer de duplication de la donnée.

La figure 6.13 illustre ces possibilités de contextualisation de format et de valeur, selon un exemple librement inspiré d'Orchestra Networks.

Historisation des données

La solution de gestion des données de référence supporte la dernière version de la donnée mais enregistre aussi les versions précédentes.

L'historisation des données est une fonction qui enregistre chaque instance de donnée à chaque modification. Cet enregistrement est réalisé sous le même mode de stockage que le référentiel portant les valeurs actuelles (dans une base de donnée si le mode de stockage est un SGBD).

Il est ainsi possible :

- d'auditer rapidement les actions sur la donnée ;
- de restituer la donnée dans un état précédent en raison des contraintes de resynchronisation ou pour des processus consommateurs longs.

1. Voir notamment les fonctions offertes par EBX Platform de l'éditeur Orchestra Networks, dans ce domaine : http://www.orchestranetworks.com/fr/product/features_adaptation.cfm

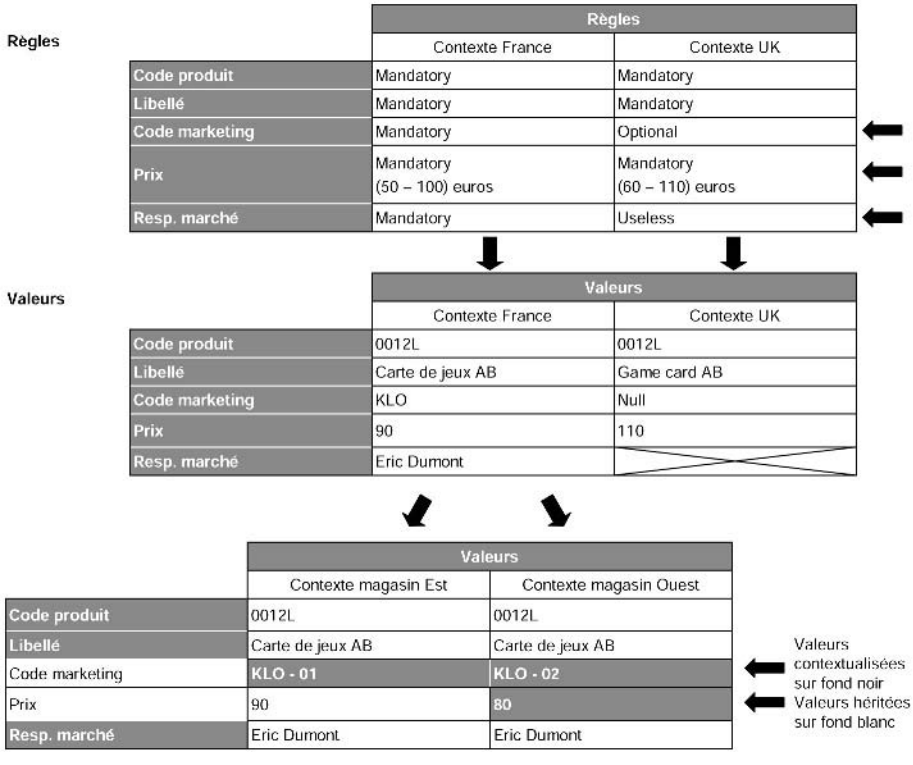

Figure 6.13 – Contextualisation de règles et de valeurs

Prenons l'exemple d'un outil de gestion de contrat d'exploitation pétrolier s'appuyant sur un référentiel de paramètres contenant les tables pays ISO (norme ISO 3166). S'il s'agit des gisements off-shore du Timor oriental, alors le code ISO alpha-2 actuel de ce pays est TL ; mais avant le 20 mai 2002, il avait pour valeur TP. Ainsi vos transactions longues, comme le versement de royalties d'exploitation, doivent bénéficier d'une historisation des données si vous désirez pouvoir les comparer dans le temps[1].

Journalisation des données

La journalisation est la capacité de la solution à enregistrer au sein d'un journal certaines actions, humaines ou automatiques, pouvant être d'intérêt dans le cadre d'une analyse a posteriori.

Cette fonction permet des analyses en cas d'erreur. Elle permet aussi d'outiller des procédures d'audit régulièrement mises en œuvre autour d'un référentiel de données confidentielles ou à risque.

1. Voir le site de l'ISO concernant cet exemple : http://www.iso.org/iso/fr/country_codes/updates_on_iso_3166.htm

6.1.6 Fonction d'accès et de diffusion des données

Contextualisation et règles de visibilité

Il est possible de définir plusieurs contextes dans les produits de MDM (par exemple, le prix d'un produit qui varie selon les pays et qui est exprimé dans une devise différente). L'utilisateur qui se connecte ne doit voir que les attributs et les règles qui le concernent.

La gestion des accès répond à une gestion fine des droits (voir la section « Gestion des droits » ci-après).

On identifie cependant différents modes de consommation ayant un impact sur la gestion de la visibilité des objets. La différence est notable si c'est une personne ou une machine qui interroge le référentiel.

Pour la consultation par IHM de l'information, les écrans s'appuient directement sur les droits gérés par la solution.

Pour les services de consultation en mode requête (services Web par exemple), on peut aussi appliquer un profil lors de la mise à disposition de l'information.

Cependant, on préfère souvent, lors de l'usage d'un outil d'intermédiation entre le référentiel et des applications consommatrices, que celui-ci prenne le filtrage des données à sa charge (en mode requête ou en mode *pull*).

Le choix entre gestion au niveau du service ou filtrage par l'intermédiation dépend du niveau de confidentialité de l'information transmise.

Liberté laissée à l'utilisateur

On apprécie les solutions qui offrent à l'utilisateur la possibilité de gérer lui-même son affichage (en mode consultation ou résultat de recherche), afin de cibler les informations essentielles. Par exemple, il peut gérer l'apparition ou non de tel attribut ou telle colonne, leur ordre d'affichage, la sélection des attributs pour l'édition ; ces préférences pouvant être enregistrées sous son profil.

Recherche et filtre

Les capacités de recherche favorisent la gestion du référentiel. Elles sont aussi fort appréciées des utilisateurs métier qui peuvent définir des sous-groupes d'objets afin de pourvoir ensuite les éditer et les utiliser au sein d'applications bureautiques, par exemple.

La recherche doit pouvoir être multicritère, mais limitée par la gestion des droits. Par exemple, les utilisateurs de base peuvent voir le chiffre d'affaires généré par chaque client, mais n'ont pas le droit de recherche sur la valeur du CA afin d'identifier les meilleurs clients (cette fonction étant réservée aux commerciaux seniors).

La nuance entre **recherche** et **filtre** porte sur la nature des recherches possibles. Le filtrage se limite à conserver les objets répondant aux critères de valeurs affectés à chaque attribut lors de la recherche. Tandis qu'une fonction de recherche complexe

doit pouvoir remonter les liens et relations entre les objets afin de dresser une liste (par exemple, la liste des membres d'une cellule familiale comportant au moins deux enfants).

On appréciera les solutions qui permettent d'enregistrer des recherches afin de pouvoir les rejouer d'un simple clic, ou encore de pouvoir les dupliquer pour créer des variantes.

Les recherches au sein du référentiel peuvent être très consommatrices de ressources et demander des ajustements de l'application lors de sa mise en œuvre. Ainsi, dans IBM MDM Server, il est possible de paramétrer un attribut en le rendant « searchable » afin d'améliorer les performances de recherche sur celui-ci. D'autres outils obligent à créer ses propres tables d'indexation.

Edition

La fonction d'édition permet la création de fichiers en formats standard directement utilisables par l'utilisateur ou pour l'extraction ponctuelle de données vers des partenaires. Ainsi, suite à une recherche et/ou un filtrage, on a la possibilité de générer des fichiers TXT, XLS ou encore XML portant sur le résultat. On veillera cependant à limiter les droits d'édition de certains référentiels.

Prenons l'exemple d'un référentiel client : si chaque fiche client n'est pas confidentielle en soit, l'extraction de l'ensemble de la base client peut cependant représenter un risque. On pourra limiter la capacité d'édition de tel ou tel attribut, alors que ces mêmes attributs restent visibles grâce aux IHM de la solution en Intranet (limitation des droits par fonctions).

Gestion événementielle

La gestion événementielle permet au référentiel de déclencher des actions sur des événements advenus. Ces événements peuvent assurer une continuité transactionnelle suite à l'arrivée d'un objet, comme la récupération successive d'informations depuis les sources ou le déclenchement d'une mise à disposition vers les cibles. Ils peuvent aussi être ordonnancés (*scheduling*) ou encore intervenir sur une vérification programmée (action sur atteinte de seuil ou de date par exemple).

Si la maîtrise de ces événements importe autant pour la synchronisation amont (complétion auprès de toutes les sources) que pour la synchronisation aval (alimentation des cibles), elle doit aussi permettre de déclencher des actions de pilotage comme des alertes ou déclenchement de flux pour mise à jour sur des informations obsolescentes.

6.1.7 Administration et maintenance

Gestion des droits

Les droits sur les processus référentiels, notamment les actions de chaque utilisateur, répondent à une matrice de propriété selon *N* axes. Les utilisateurs, les applications

ou même les SI partenaires peuvent bénéficier de droits limités à un périmètre restreint de la donnée.

Les restrictions de droits peuvent s'appliquer soit en « largeur » soit en « profondeur ». En largeur, on limitera l'accès à un sous-ensemble d'instances d'objets détenus par le référentiel (par exemple, accès aux clients du segment « 30-45 ans »). En profondeur, on limitera l'accès à tel ou tel attribut du modèle de données. Pour ce faire, on remarque que la gestion des droits doit pouvoir s'appuyer :

- sur une gestion au niveau du modèle (tel rôle peut faire telle chose sur tel attribut) ;
- mais aussi sur une gestion au niveau du contenu des instances d'objet (tel rôle a tel droit car telle valeur est portée par l'objet).

La figure 6.14 schématise cette gestion des droits pour un produit représenté selon une matrice à *N* dimensions (ici 3), suivant les axes :

- cycle de vie (les droits peuvent être différents selon l'état métier du produit) ;
- attributs (tous les attributs ne sont pas disponibles pour chacun) ;
- rôles utilisateurs.

Au croisement des *N* dimension se définissent les droits alloués (par exemple, création, modification, validation, suppression…).

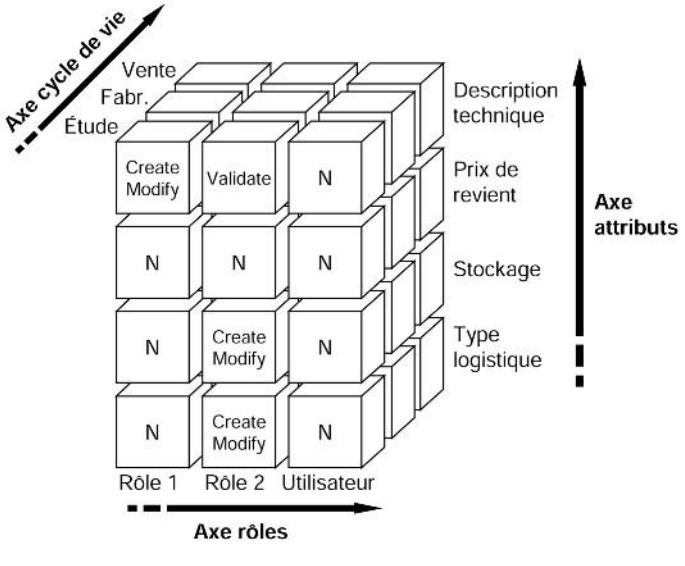

Figure 6.14 – Gestion des droits selon plusieurs axes

On différencie la gestion des droits de celles de l'authentification. Dans un SI sécurisé, on intègre la solution de MDM aux protocoles de sécurité du SI, en le con-

nectant notamment aux annuaires d'authentification LDAP et au SSO (*Single Sign On*) le cas échéant. Les droits, quant à eux, restent dans le périmètre de gestion de la solution MDM.

Versionning

Le modèle de la donnée est appelé à évoluer au cours du temps (attributs, structure) pour répondre aux évolutions du métier ou des applications. La solution de gestion de données de référence doit permettre la gestion simultanée des versions concourantes des objets supports d'une même donnée.

Ce point est généralement bien supporté en ce qui concerne les évolutions de structure (simple extension du modèle de donnée comme l'ajout d'un attribut). Dans le cas d'une modification de la structure de l'objet, la réponse est généralement plus lourde et demande la gestion d'instances concourantes du référentiel, grâce à la couche d'intermédiation.

Contrôle de synchronisation

Le référentiel définit le point de vérité de la donnée. Les applications consommatrices se réfèrent à ce point de vérité pour leurs propres données internes. Mais quelle en est la valeur s'il est impossible de garantir que les données utilisées dans les applications consommatrices sont bien identiques aux données validées au sein du référentiel ?

Un contrôle de synchronisation entre les applications consommatrices et le référentiel permet d'assurer cette validité.

Ce contrôle peut techniquement être opéré « au fil de l'eau » ou en mode *batch* :

- « au fil de l'eau » : c'est le mode le plus simple et le plus pratique ; il s'appuie sur la génération d'un identifiant de synchronisation ou identifiant d'instance de donnée (réactualisé à chaque création ou modification d'une donnée d'un objet ou à chaque mise à disposition pour tel ou tel groupe de consommation). Ce code est capitalisé par les outils d'intermédiation. En mode allégé, ce code est retourné lors de la « bonne livraison » à l'application consommatrice. En mode complet, ce code est retourné dans un acquittement fonctionnel par l'application validant ainsi la « bonne intégration » de la donnée. Le contrôle de synchronisation est ainsi opéré au niveau des outils d'intermédiation et de leur outil de supervision fonctionnel ;
- mode *batch* : il convient pour chaque identifiant détenu par l'application consommatrice d'extraire les champs critiques de l'objet et de les comparer à ceux détenus par le référentiel. Cette comparaison est à la charge des outils d'intermédiation et de leur outil de supervision fonctionnel.

6.2 LES CATÉGORIES DE SOLUTIONS MDM

Les applications MDM se déclinent en plusieurs catégories. Ces catégories dépendent du type de données supportées à l'origine par les applications et de la spécialité des éditeurs des solutions.

Ainsi, les solutions de type catalogue pour le support des référentiels *produit* sont les premières apparues sur le marché, suivies par les solutions orientées *clients*. Ces solutions ont été massivement rachetées par les grands éditeurs métier ou techniques de la place (Trigo et DWL racheté par IBM, A2I racheté par SAP…). Les éditeurs BI ont, quant à eux, travaillé sur les solutions de *key mapping* et enfin des « Pure Player » ont édité leurs solutions dédiées au MDM.

Nous avons donc quatre catégories principales d'outils :

- PIM, pour *Product Information Management* ;
- CDI, pour *Customer Data Integration* ;
- *key mapper*, issus du décisionnel ;
- génériques, conçus dès le départ pour supporter tous types de données de référence.

Historiquement, les solutions PIM offrent des capacités collaboratives (*workflow* simple), initialement conçues pour créer et valider les *produits* et *catalogues*. Ces solutions offrent toutes la capacité de modéliser à volonté n'importe quel objet. Cette modélisation est une modélisation logique, le modèle physique n'étant pas accessible à l'administrateur.

Les solutions CDI sont généralement des solutions de nature transactionnelle, plutôt conçues pour fonctionner en architecture de coopération. Elles intègrent des capacités DQM ou des sources expertes externes.

Les *key mapper* sont orientés pour la gestion des clefs et des hiérarchies ; leurs fonctionnalités sont généralement limitées autour de ce périmètre.

Les génériques sont peu nombreux. Ils offrent des solutions flexibles, pouvant intégrer tous types de modèle de données (généralement en XML/XSD). Les meilleurs outils de cette catégorie proposent maintenant un large éventail de fonctions tel que décrit précédemment. Ainsi, on retrouve *workflow* simple, gestion de hiérarchies, contextualisation, version de référentiel…

L'offre sur le marché international est pléthorique et elle le reste sur le marché français. La figure 6.15 illustre cette profusion. Nous avons classé les génériques et les *key mappers* ensemble pour une meilleure lisibilité et car ils sont amenés à traiter tous types de donnée par nature.

La taille des bulles n'est pas signifiante en termes de part de marché, nous avons tenté de symboliser l'étendue que peuvent couvrir les solutions en termes de modèle. Cela ne pouvant être réalisé pour chaque solution, nous avons limité ce mode de représentation à quelques-unes.

Cette représentation est un instantané du marché et la pérennité de cette figure n'est que de courte durée.

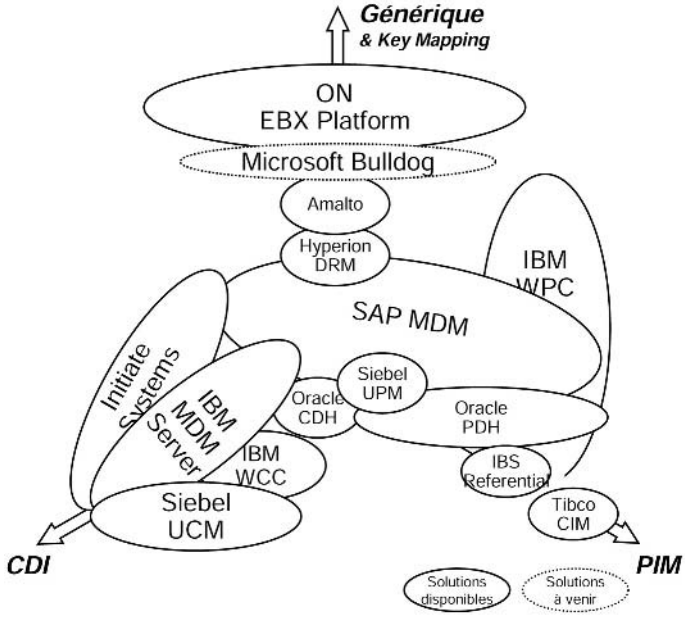

Figure 6.15 — Offre par catégorie de solution

Enfin, signalons que cette catégorisation tend à s'estomper tandis que le marché devient mature. Chaque éditeur cherche à faire converger ses solutions afin d'offrir une plate-forme de gestion unifiée.

6.3 SOCLE RÉFÉRENTIEL

La notion de socle référentiel recouvre l'ensemble des services et applications mutualisés entre les référentiels. Cette notion est essentielle dans le cadre d'une démarche générale ou multiréférentiel.

On pourra subdiviser les couches de ce socle entre services référentiels partagés et applications transverses.

Pour les services référentiels partagés, on se réfère aux éléments – couverts dans ce chapitre – de la description fonctionnelle d'une solution de gestion des données de référence. Cette partie du socle est couverte par des développements spécifiques (générateur d'identifiant par exemple) ou par des applications propres à la gestion des données et pouvant être partagées entre les différents référentiels (DQM, indicateurs, glossaire métier, *metadata repositories*…). On notera la nature particulière du

« référentiel de paramètres » qui est partagé entre tous les référentiels et qui est lui-même un référentiel utilisé par les *middlewares* ou les applications du SI.

Pour les applications transverses, on se référera aux applications partagées du SI, par exemple EAI et ETL pour les échanges, LDAP pour l'authentification, Portail pour l'intégration IHM…

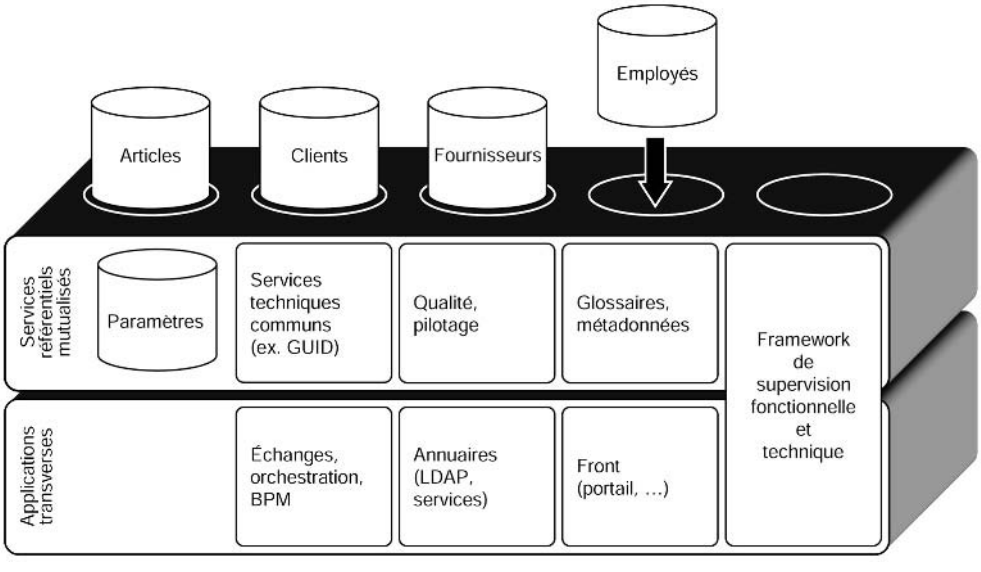

Figure 6.16 — Le socle référentiel

La constitution de la couche « applications transverses » n'est pas propre aux projets MDM. Le MDM s'intègre donc au SI dans l'état qui est le sien. On soulignera cependant l'importance de la couche d'échange (voir chapitre suivant).

La constitution de la couche « services référentiels partagés » se fait par étapes (à chaque itération d'un projet). Il faut commencer par les services techniques communs et la mise en place d'une supervision, puis procéder à l'ajout d'outils de pilotage et de gestion des métadonnées.

L'importance de la cohérence technologique entre les solutions doit être soulignée. Mais la cohérence technologique (Java, XML, SOA…) n'implique pas obligatoirement l'utilisation de la gamme d'un unique éditeur pour couvrir tous les besoins. Il reste essentiel de répondre d'abord aux besoins métier et donc une approche « *best of breed* » (meilleur produit par rapport à une fonction) peut être envisagée.

En résumé

Les solutions de gestion des données de référence se doivent d'offrir de nombreuses fonctions de gestion et de gouvernance : acquisition de la donnée, validation et qualité, pilotage, modèles, stockage et journalisation, diffusion, administration.

Le panel des fonctions offertes dépend en particulier de la typologie d'outil MDM que vous implémentez (CDI, PIM, génériques).

Certaines de ces fonctions peuvent être mutualisées entre les référentiels et constituer un « socle référentiel ».

7

Positionner le référentiel dans le SI

Objectif

Après avoir exposé les grands types d'architecture ainsi que les fonctionnalités attendues pour une solution MDM, nous allons évoquer l'intégration et la place du référentiel au sein du SI de l'entreprise.

Cette intégration répond aux principaux objectifs évoqués dans la première partie. Elle repose aussi sur une intégration étagée afin d'éviter un mode « big-bang » préjudiciable à la réussite des projets.

La place du référentiel, son interaction avec l'ensemble des couches applicatives du SI et le bon usage de la couche d'intermédiation sont l'objet central de ce chapitre.

7.1 BRIQUES APPLICATIVES

Dans la conception des applications informatiques, on considère cinq grandes couches représentées sur la figure 7.1 :

- accès (portail, B2B…) ;
- intermédiation (EAI, ETL…) nécessaire pour des échanges entre machines ;
- métier (logique applicative construite autour de progiciels, serveurs d'applications, serveurs de données…) ;

- pilotage (décisionnel, BAM, supervision technique) ;
- transverse (sécurité, annuaire, pilotage).

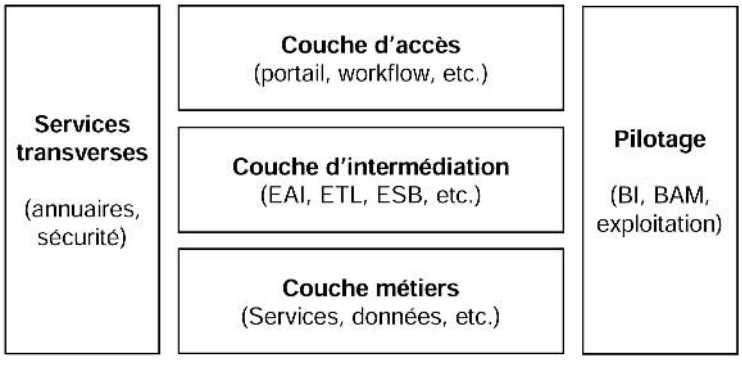

Figure 7.1 — Les cinq grandes couches applicatives

Le schéma de la figure 7.2 présente dans le détail les domaines fonctionnels (regroupement de besoins fonctionnels, comme la gestion de contenu) et les briques fonctionnelles (en général associées à un type de besoin). On voit apparaître un domaine gestion des données, dans lequel on trouve les produits que nous avons déjà évoqués : MDM (*Master Data Management*) et DQM (*Data Quality Management*). L'EII (*Entreprise Information Integration*) est, quant à lui, cantonné à l'intermédiation. Par ailleurs, on note les domaines fonctionnels suivants qui seront abordés dans les architectures présentées :

- **Gestion applicative** : il s'agit des progiciels métier, mais aussi des briques techniques supports ou constituantes d'une solution spécifique (les serveurs Web, les serveurs d'applications qui permettent en particulier de construire des applications transactionnelles et de manière plus générale tout type de traitement).
- **Échanges et orchestration** : il s'agit de la couche d'intermédiation avec les technologies telles que l'EAI (*Enterprise Application Integration*) pour l'échange de messages à travers un bus, l'ESB (*Enterprise Service Bus*) pour l'appel aux services Web à travers un bus, l'ETL (*Extract Transform Load*) pour le chargement en masse de données avec transformations de modèles, le BPM (*Business Process Management*) pour la définition et l'activation des règles d'enchaînements de traitements (le *workflow* étant l'enchaînement de tâches réalisées par des humains). Il s'agit aussi des moteurs de règles qui permettent de définir et de modifier les règles d'enchaînements en dehors des applications (mais aussi de coder les algorithmes de règles métier complexes).
- **Accès et gestion de contenu** : ils constituent la couche d'accès aux informations, en général par les humains mais aussi par les applications notamment via le B2B (*Business to Business*). Elle permet à la fois de manipuler des

données structurées (typiquement stockées dans des SGBD) et des données non structurées (documents aux formats divers rangés dans des fichiers).
Remarque : la gestion de contenu est actuellement peu concernée par les solutions MDM (simple gestion de cohérence), mais l'extension de l'EIM (*Entreprise Information Management*) aura un impact sur cette sphère. Il peut s'agir par exemple de la réglementation pour un suivi de certificat lié à l'écologie ou à la traçabilité sanitaire et plus tard pour la mise à disposition d'information ciblée non structurée provenant de sources de confiance, tels que les éditeurs de contenus.

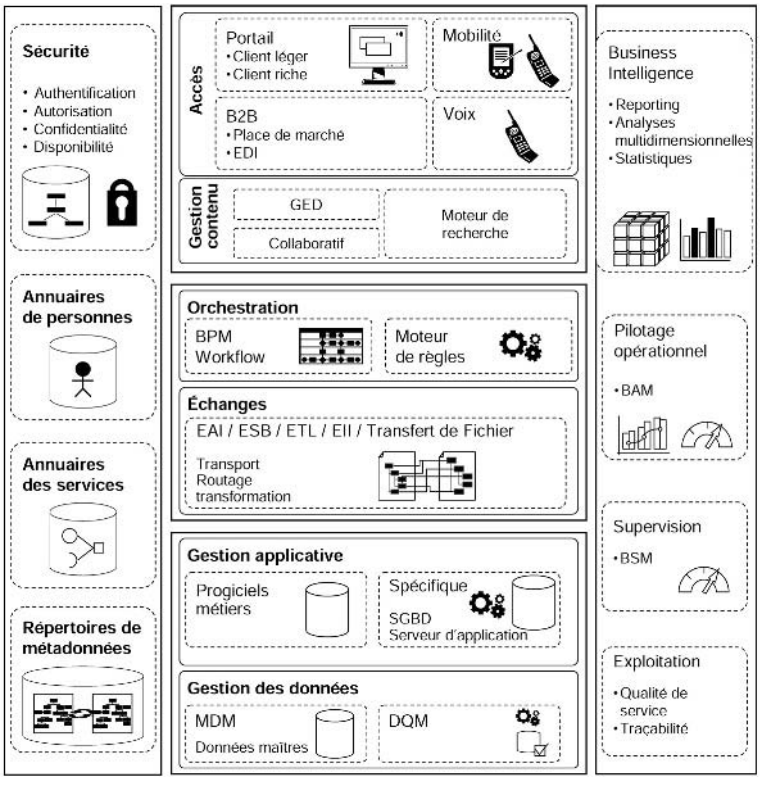

Figure 7.2 — Les principales briques applicatives

- **Pilotage** : il englobe la *Business Intelligence* (BI) ou le décisionnel, le BAM (*Business Access Management*), le BSM (*Business Services Management*) et les outils de supervision applicative. La BI permet de construire des tableaux de bord, des rapports d'analyse, des prévisions… Cette brique fonctionnelle est fortement liée à une bonne gestion des données de référence car la qualité des données induit des rapports ou tableaux de bord reflétant réellement l'activité et les performances de l'entreprise. Une politique de bonne mise en œuvre de la BI ne peut pas ignorer deux sujets que nous évoquons largement dans ce

livre : la gouvernance des données (essentielle pour obtenir des données de qualité), la mise en œuvre éventuelle d'outils de gestion de la qualité des données et la gestion adéquate des données de référence. Le BAM, qui permet un *reporting* sur le bon fonctionnement des processus métier, est plus marginal par rapport à notre propos.

- **Sécurité** : c'est un sujet important, qui concerne à la fois l'authentification (prouver son identité), l'habilitation (droits pour réaliser des opérations), la confidentialité (chiffrement des données confidentielles si nécessaire), la disponibilité (sauvegardes, reprises après pannes, dispositifs de haute disponibilité…). C'est un domaine par ailleurs largement outillé et pris en compte par les DSI des entreprises que nous évoquerons peu dans ce livre.
- **Annuaires** : on distingue les annuaires de personnes (de type LDAP), les annuaires de services (au sens des services Web) et enfin les annuaires de métadonnées (déjà évoqués et sur lesquels nous reviendrons). Les technologies d'annuaire peuvent être considérées comme des solutions de gestion de certaines données de référence comme, par exemple, les employés d'une société. C'est pourquoi nous les avons évoquées dans les solutions possibles.

Les services Web ne figurent pas sur ce schéma en tant que tels ; ce sont des technologies utilisables en développement pour construire des architectures SOA (voir l'annexe sur ces architectures). De manière très simplifiée, les architectures SOA articulent les applications et le SI plus généralement, autour de services qui sont des entités d'exécution autonomes accessibles via une interface publiée. À noter aussi qu'un progiciel métier (type SIEBEL ou SAP CRM) possède ses propres briques (par exemple, SAP PI est un EAI, SAP Portal un portail) en plus d'une puissante composante métier spécifique généralement bâtie aussi autour d'un serveur Web et d'un serveur d'application.

Conséquemment aux chapitres sur l'architecture fonctionnelle, on notera qu'une solution de gestion de données de référence utilise potentiellement de nombreuses briques applicatives issues du modèle ci-dessus. La mise en œuvre par étapes et une analyse propre des besoins est donc essentielle pour doser l'effort de chaque itération projet et éviter un « big bang » insurmontable.

7.2 PROJETS CLASSIQUES ET LEURS APPLICATIONS

À partir des modèles présentés dans la section précédente, il est possible de définir des architectures types très simplifiées pour un ensemble de projets classiques. Nous proposons ainsi trois types de projets pour lesquels nous déclinons l'architecture applicative :

- amélioration des processus ;
- échange et B2B ;
- analytique.

Notre intention est d'identifier les écarts et apports entre une architecture classique et une architecture MDM lors de la mise en œuvre de tels projets.

7.2.1 Amélioration de la performance des processus

L'amélioration de la performance des processus (figure 7.3) met en œuvre l'automatisation orchestrée d'un processus intra-SI, au sein duquel interviennent des acteurs humains et des applications. Les applications peuvent être légataires (*Legacy*), et l'interface utilisateurs se fait dans le cadre d'IHM agrégées de type portail (ou client riche sur le poste de travail), voire sur des terminaux mobiles.

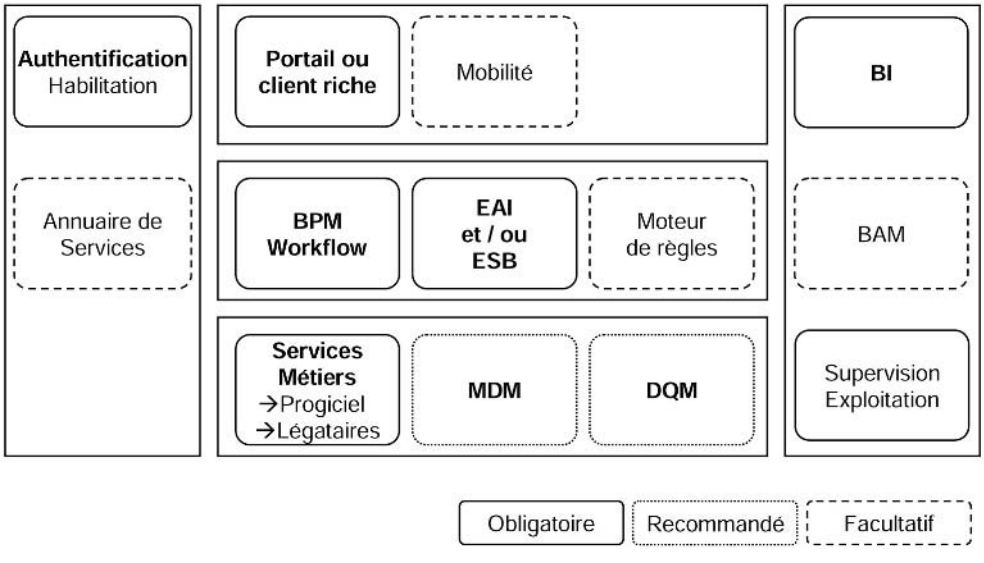

Figure 7.3 — Les principales briques applicatives pour améliorer la performance des processus

Plus le processus est transverse et important, plus le MDM est générateur de valeur.

Prenons un processus transverse tel que « traiter une commande » qui repose sur les données de références métier produit, fournisseur, client, organisation (site logistique par exemple), personne (pour les commerciaux par exemple) et sur de nombreuses données de paramètre (objet d'imputation, devise…).

L'analyse de la mise en œuvre du MDM se fait donnée par donnée (éligibilité du MDM) et pour chaque donnée, attribut par attribut (éligibilité au référentiel).

Cette analyse répond à de multiples considérations issues du cadre de gouvernance (voir troisième partie). Elle se fait au périmètre du projet concerné pour les critères principaux mais aussi suivant des enjeux plus larges (portefeuille projet, stratégie SI…) pour des critères secondaires. La prise en compte et la pondération des critères tant primaires que secondaires doit s'opérer pour tous les projets d'importance car elles sont généralement à l'origine d'un chantier MDM au sein des grands projets. Ainsi, le déclenchement d'une démarche outillée de gouvernance des données pourra apparaître de manière opportuniste et tactique sous l'impulsion des urbanistes et de la DSI.

Premier zoom sur le cadre d'analyse

Pour chaque donnée, on analyse les contraintes et objectifs issus du cadre de gouvernance (voir partie 3), comme indiqué ci-après :

La stratégie d'entreprise : si l'entreprise est dans une phase de croissance externe, on favorise par exemple l'outillage des données descriptives et constitutives de la nouvelle organisation. On considère en priorité les référentiels Employés et Organisation pour la partie descriptive mais aussi les référentiels de *back office* pour la partie constitutive, comme le référentiel de structures comptables afin de permettre la consolidation des résultats et la création des liasses fiscales.

La conformité réglementaire : il s'agit par exemple de la confidentialité des informations nominatives et des traitements associés (en rapport avec la CNIL), de la traçabilité en matière écologique ou sanitaire… Le MDM en tant que point focal permet un contrôle accru ou une meilleure indépendance entre entité et processus (séparation entre commercialisateur et distributeur comme l'impose la Commission de régulation de l'énergie, par exemple).

La qualité : voir les niveaux de qualité intrinsèques et de services évoqués dans la première partie.

La sécurité : le MDM peut être utilisé en réponse aux risques relatifs à la donnée (risque financier ou en termes d'image).

On analysera aussi les axes de mise en œuvre influençant l'architecture, exposés ci-après :

Les métiers et l'organisation : ils sont importants afin de valider l'adhérence des sous-processus amont et aval aux référentiels, et d'identifier les acteurs de ces processus. On identifiera ainsi si le référentiel doit être considéré comme un référentiel technique (à la charge de la DSI) ou un référentiel fonctionnel (à la charge des métiers).

Plus le référentiel est fonctionnel (besoins de contrôle, nombre de participants), plus il embarque des fonctions propres au pilotage et favorise donc le MDM. Complétée par le périmètre de donnée gérée, cette étude permet de déduire le mode d'implémentation.

L'urbanisme : en première analyse, on souligne l'importance de :

- la dispersion de la donnée (partage du paradigme) entre processus propres au projet mais aussi avec les autres grands processus de l'entreprise ;

- la volatilité de la donnée (temps moyens entre modification d'instance, et nombre d'instances d'objets modifiées sur l'ensemble des instances d'objets détenues par un même paradigme).

La dispersion s'analyse en amont et aval du référentiel. En amont, il faut choisir entre les principaux types d'architecture (voir la section 4.2) mais aussi d'outillage. En aval, apparaissent des besoins en termes de normalisation, gestion événementielle et diffusion.

L'analyse répond à la maîtrise du portefeuille projet (et donc de la pérennité des applications dans le SI) afin d'identifier les périmètres de données et le mode d'implémentation à préférer. La dispersion commande ainsi le Plan de transformation du référentiel, c'est-à-dire l'évolution par étape du référentiel (périmètre et type d'implémentation, donc fonctions) et la maîtrise des flux entrants et sortants.

La volatilité de la donnée induit une architecture agile, une intermédiation rapide (message en synchrone ou « au fil de l'eau ») et un contrôle accru du synchronisme.

La conduite du changement : il s'agit d'outiller directement la solution avec des outils de supports tel que l'accès direct au dictionnaire sémantique, des aides en ligne ou un contrôle de saisie.

Les méthodes et outils : on analyse les méthodes et outils existants ainsi que ceux du portefeuille projet.

Au périmètre du MDM, on analyse les différents scénarios de mise en œuvre et le périmètre des fonctions attendues en les confrontant aux autres dimensions du cadre. On en déduira un plan projet en accord avec les charges, le temps et le budget disponibles. On se pose notamment la question de l'utilisation d'un progiciel de gestion comme référentiel, le développement d'un logiciel spécifique ou l'utilisation d'un référentiel MDM.

Application dans un exemple concret

Reprenons notre exemple sur le processus « traiter une commande ». Notre entreprise est un industriel, ses clients sont des distributeurs spécialisés répartis dans le monde entier. Les équipes commerciales gèrent l'ensemble de leurs activités pendant leurs déplacements clients. L'entreprise est stable et ne prévoit pas de procéder à un changement de périmètre (rachat/fusion) dans l'immédiat. Le DSI se souvient pourtant des impacts apparus lors de la dernière réorganisation interne. L'objectif de l'entreprise est d'améliorer les délais de traitement des commandes, des factures et de leur recouvrement.

Force de vente, logistique et comptabilité sont les trois activités principales concernées par la refonte du processus. La force de vente doit pouvoir gérer en mode connecté ou non connecté ses offres, ses clients et ses commandes. La logistique doit améliorer sa réactivité par une meilleure maîtrise des stocks et des expéditions. La comptabilité doit automatiser l'édition et la relance des factures, proposer une auto-

matisation complète à ses principaux clients et améliorer la maîtrise des crédits clients.

L'ensemble des processus dispose d'indicateurs de pilotage pour en apprécier la performance.

L'outillage métier prévu repose sur un progiciel CRM (orienté SFA, *Sales Forces Automation*) pour la force de vente et un second progiciel de gestion pour la logistique et la comptabilité (PGI). Le SI dispose déjà d'outils d'urbanisation comme un EAI. Une première maquette SOA sur BPM a été réalisée en prévision d'une stratégie SI plus générale. Les processus transverses sont outillés par BPM et orchestrateur. Un portail de e-commerce existe déjà et ne sera que peu modifié car il répond aux besoins clients mais l'ensemble des flux d'alimentation sera revu.

Concernant les données de référence, les catalogues produit et les offres sont gérés dans le frontal Force de vente, offrant une déclinaison locale pour chaque pays. Les stocks sont rafraîchis « au fil de l'eau » depuis le progiciel de gestion dans le CRM. Les données clients sont acquises principalement par le frontal Force de vente et complétées dans le progiciel de gestion par la logistique et la comptabilité. L'organisation est décrite dans le progiciel de gestion, seuls les commerciaux sont identifiés dans l'outil de CRM/SFA.

Après analyse, les choix suivants sont établis :

- Gestion des produits et offres maintenue dans le CRM/SFA avec flux de synchronisation entre SFA et le PGI. Le CRM/SFA offre un taux de couverture des besoins du PGI proche des 100 %, les adaptations en « spécifique » sont mineures.
- L'acquisition des clients est réalisée dans le CRM/SFA et le PGI, avec comme particularité que le SFA est seul apte à « créer », le PGI ne peut que « compléter » une donnée qu'elle reçoit (cela simplifie l'orchestration des synchronisations amont). Un référentiel Client, suivant une architecture de coopération, est prévu. La force de vente n'a ainsi qu'un unique outil gérant le mode connecté ou non.
- Le référentiel offre des capacités DQM pour s'assurer de l'unicité et de la normalisation stricte des données. Ces capacités sont aussi utilisées lors de la reprise afin de constituer le référentiel initial à partir des sources des différents pays. Le chargement initial du PGI et du SFA est réalisé par extraction du référentiel, puis les flux amont entre SFA, PGI et référentiel sont déployés. Le DQM outille aussi les indicateurs de pilotage du référentiel.
- Le référentiel offre la possibilité de reconstituer les hiérarchies clients ; ainsi la vue des groupes est possible. Cette aptitude offre, *via* un développement spécifique adossé au PGI, la possibilité de mieux gérer les encours de crédit. En effet, la somme des crédits possibles alloués aux différents établissements de vente peut dépasser celle allouée à la branche régionale ou au groupe en entier. La vision reconstituée des filiations permet de refuser un crédit à un établissement, non pas parce que son propre encours possible est dépassé, mais parce que celui de sa maison mère l'est.

- Un logiciel LDAP spécifique est alimenté par le référentiel client afin d'ouvrir les droits à l'automatisation des transactions financières.
- Le référentiel propose, en outre, des capacités de recherche et d'édition au travers du portail d'entreprise sous client léger.
- Les flux amonts sont outillés en ESB, une partie des processus est portée en BPM, les flux aval utilisent EAI, ESB et ETL en fonction de la cible.
- L'organisation et les sites sont modélisés et gérés dans le PGI au périmètre du projet. Le DSI préfère envisager un futur projet de refonte du SI Ressources humaines pour créer un véritable référentiel Organisation et Employés.
- Les urbanistes désiraient se doter d'un référentiel de paramètres, mais malgré les risques concernant la transcodification et sous la pression des métiers et de l'équipe BI, ils ont dû renoncer. Chaque table de paramètre est acquise et gérée dans les applications, l'intermédiation est mise à jour manuellement pour les transcodifications. Le BI assure l'historisation des tables et possède sa propre gestion des transcodifications.

On aboutit au SI illustré par la figure 7.4, tandis que la situation idéale du point de vue gestion des données de références est donnée à la figure 7.5.

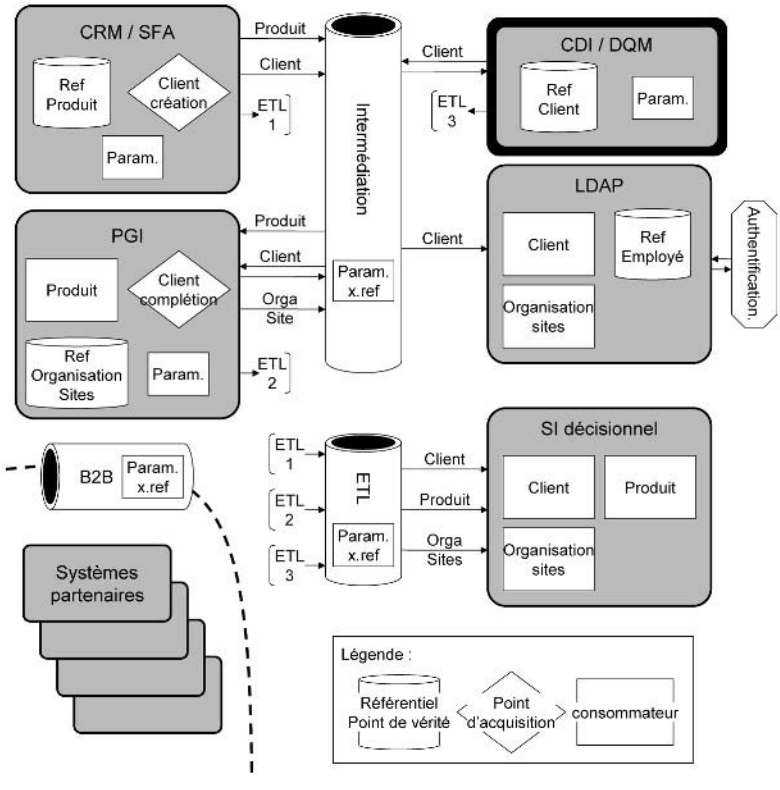

Figure 7.4 — SI défini par l'analyse pour outillage du processus « *Traiter une commande* »

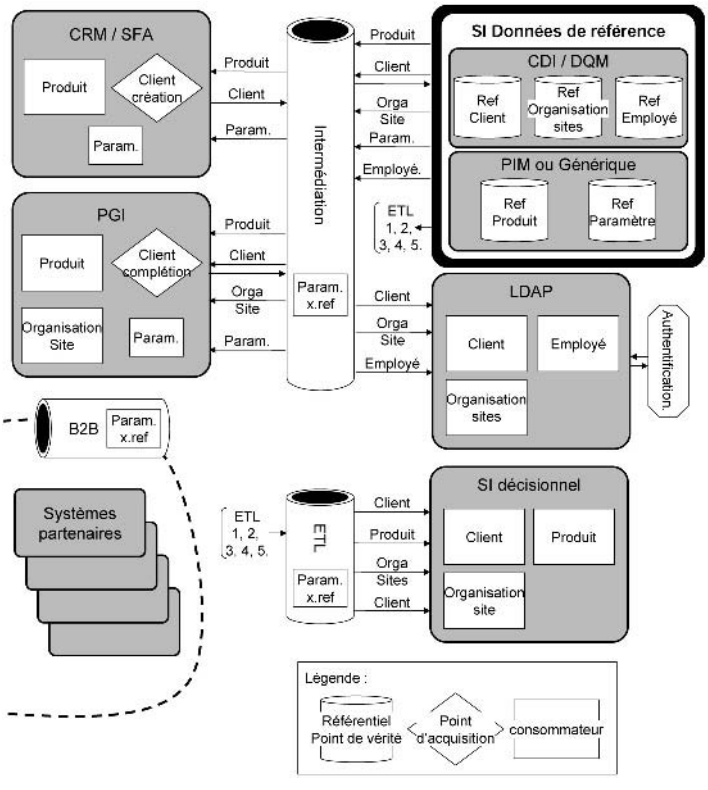

Figure 7.5 — SI cible défini par l'analyse pour outillage du processus « *Order to Cash* »

7.2.2 Interfaçage du SI avec les tiers (communication B2B)

Cette typologie de projet concerne la mise à disposition de fonctions ou d'informations gérées au sein d'un SI, soit à des utilisateurs humains, soit à des SI automatisés externes.

Les conditions de sécurité des accès tiennent un rôle important dans cette architecture.

Annuaire de services et BPM ont un intérêt s'il y a partage de processus entre l'entreprise et ses partenaires. Le BI est lui aussi facultatif pour des raisons semblables, il ne mesure que ce qui a besoin de l'être. Les moyens d'accès sont généralement multiples (EDI, portail, flux spécifiques) car les partenaires de l'entreprise présentent le plus souvent des niveaux d'informatisation très différents.

Ici encore, le MDM est une brique facultative mais recommandée. Il offre trois fonctions principales : la normalisation des données (incluant leurs règles de validité), la gestion événementielle de mise à disposition des informations en amont ou en aval et le contrôle des partenaires (droits et niveau de service de ceux-ci). Le contexte du type d'informations échangées décide de la mise en œuvre de ces fonctions et de leurs extensions. Voici deux exemples :

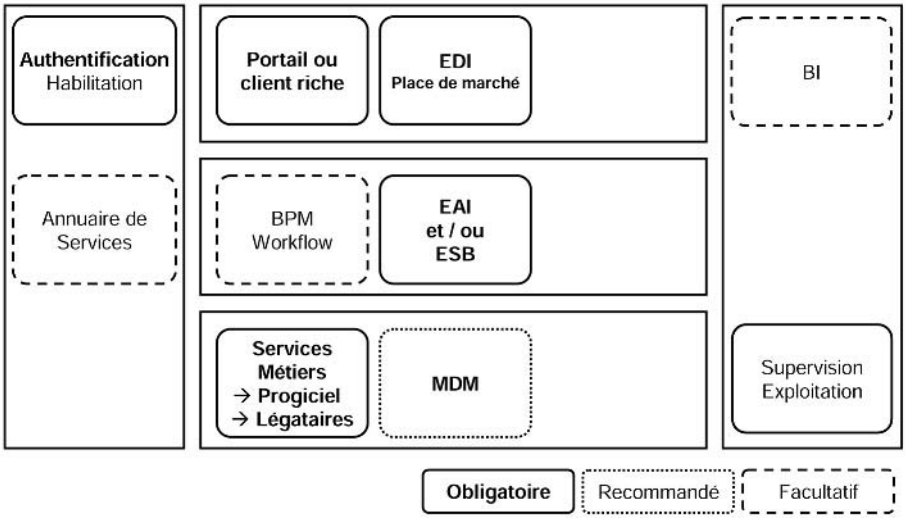

Figure 7.6 — Les principales briques applicatives pour l'interfaçage du SI avec les tiers

- Dans le cadre d'un partage de catalogue (pour les services achats ou pour le référencement dans la distribution), le MDM couvre fournisseurs et articles/produits. Il gère les droits, valide les données entrantes avant mise à disposition. Il peut capitaliser les erreurs en remontée depuis les services internes et être le point de mesure de la qualité des informations entrantes de chaque fournisseur.
- Dans un cadre réglementé, comme pour les télécommunications ou l'énergie, les gestionnaires des réseaux de distribution ne sont pas propriétaires des données clients et ne connaissent que les points de livraison et de mesure. Le référentiel client du commercialisateur et les référentiels point de livraison et point de mesure du distributeur sont utilisés pour des processus tels que « ouverture de service » ou « facturation ». Le distributeur n'ayant pas l'information client, la cohérence est opérée dans le référentiel client du commercialisateur. Ainsi, il ne s'échange que des informations anonymes et normalisées.

7.2.3 *Reporting* et analyse

Ce projet type récolte les informations des processus métier et permet de les tracer. Les transactions des applications en support de ces processus alimentent le BI en données transactionnelles pour permettre la mesure de la performance des processus qu'elles outillent. Ces données transactionnelles sont corrélées entre elles afin de permettre la création d'indicateurs et des possibilités d'analyses multidimensionnelles. La sphère décisionnelle enregistre chaque transaction et autorise une vision dans le temps.

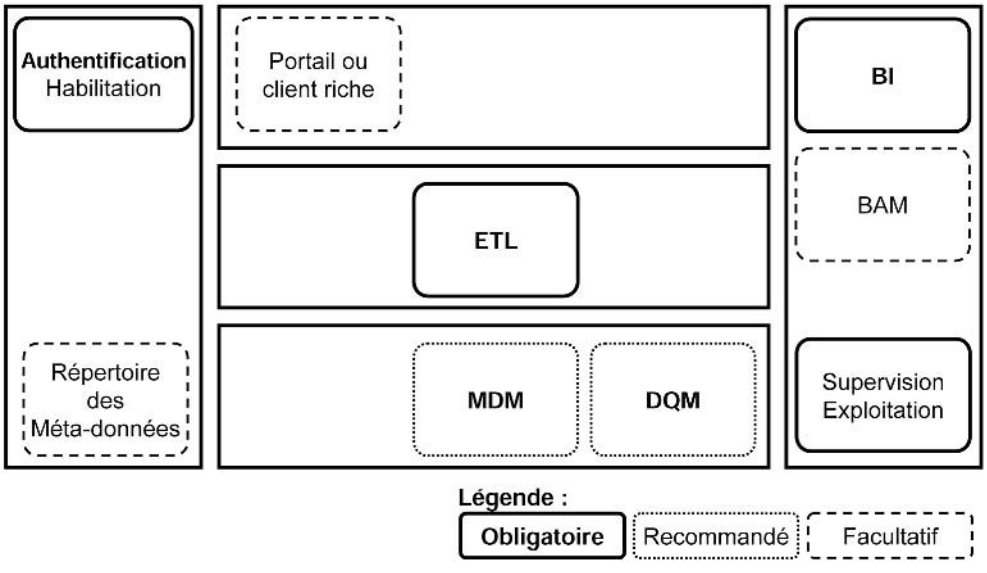

Figure 7.7 — Les principales briques applicatives pour *reporting*

Jusqu'à récemment les SI décisionnels étaient construits afin d'être les plus autonomes possibles. Placés en fin de chaîne de l'information, ils étaient dotés de tous les outils nécessaires à leur fonctionnement. Cette autonomie est maintenant entamée pour plusieurs raisons :

- les cas d'utilisation évoluent ;
- le BI génère de plus en plus d'informations réinjectées dans les outils transactionnels (calcul de segmentation, *scoring*, consolidation financière…) ;
- les bases de déversement des données transactionnelles se transforment et se doublent de véritables ODS (*Operational Data Store*) utilisés comme source de données à destination du transactionnel, notamment dans un cadre SOA ;
- les retours d'expérience du BI « indépendant » sont mitigés ;
- La qualité des données, même outillée par DQM est difficilement gérable en toute fin de chaîne (effort important pour résultat moyen). À cet égard, une maxime, que nous ne traduisons pas ici, souligne bien les problèmes que cela engendre : « *Shit in, shit out* ».

Les tables de référence des différents outils transactionnels évoluent dans le temps et demandent à être gérées pour les transcodifications. Reconstituée pour ses besoins propres et intervenant en fin de chaîne, la gestion des transcodifications est opérée en double d'autres outils dans le SI.

Les rapprochements hiérarchiques ne sont pas paramétrés une fois pour toutes, ils évoluent dans le temps eux aussi.

Le rapprochement entre objets se fait au travers de données de référence qui ne possèdent pas toujours une clef unique pour être rapprochées. Les outils DQM peuvent alors être utilisés.

La reconstitution de vue agrégée d'objets liés hiérarchiquement est parfois impossible quand ces structures sont absentes des solutions transactionnelles sources. Par exemple, pour un fichier client dont la granularité est l'adresse, il peut manquer les liens de filiation entre les instances qu'il détient et le BI est donc incapable d'opérer à ce niveau de macro-granularité.

Pour se prémunir de certaines de ces difficultés, les éditeurs BI ont créé des outils spécifiques, toujours dans une logique d'indépendance :

- il existe des outils MDM spécialisés dans le *key mapping*. On citera dans cette catégorie :
  – Hypérion, maintenant propriété d'Oracle ;
  – Stratature, maintenant propriété de Microsoft ;
  – Kalido.
- on trouve des outils DQM à l'origine des produits inclus dans les suites décisionnelles (Informatica, IBM, SAS, SAP/BO…) ;
- il existe des outils de métadonnées permettant la maîtrise des chaînes alimentant les cubes d'analyses.

Le décisionnel est à l'origine d'une approche outillée de gouvernance des données. Mais les limites organisationnelles et techniques dues à son positionnement en fin de chaîne ne permettent pas d'en tirer le meilleur parti. Une approche de gouvernance des données issue des entités décisionnelles a 90 % de chance de mal considérer les problèmes, dans une optique trop orientée. L'approche MDM se veut plus proche du transactionnel afin d'assurer une qualité au plus tôt. Cela induit un choix et une mise en œuvre spécifique des outils ainsi qu'une approche organisationnelle plus large pour impliquer et responsabiliser les métiers.

7.3 IMPORTANCE DES ÉCHANGES DE DONNÉES

Dans ces projets types, les échanges entre applications ou entre fonctions au sein d'une même application nécessitent une attention particulière. Si une architecture incluant le MDM apporte à chacun de ces projets des avantages, c'est d'abord parce qu'il repose sur un SI urbanisé au travers d'outils d'intermédiation. L'intermédiation se décline sous divers outils, protocoles, méthodes…

Allant du plus traditionnel (le « point à point ») au plus agile (la SOA), l'intermédiation est une brique applicative nécessaire au MDM. Rien ne sert d'avoir un référentiel si les données qu'il contient ne peuvent irriguer le SI.

Profitons-en pour clore un débat d'arrière-garde. La mise en place d'une couche d'intermédiation, même moderne (SOA) et respectant les meilleures pratiques, ne remplace pas une gouvernance des données, qui attribue la propriété des données à des acteurs précis et focalise celles-ci au sein d'un unique point de vérité (le référentiel).

7.3.1 Point à point

La figure 7.8 illustre un échange « point à point » entre applications. Les deux applications ont des modèles de données différents. Il faut effectuer une correspondance entre modèles et valeurs, soit dans l'application A, soit dans l'application B, soit dans l'outil d'infrastructure de l'échange. Et cela est à recommencer pour chaque échange inter-applicatif !

Sur cette figure, apparaît la notion de « modèle de flux » (voir annexe « Modélisation » pour plus de détails).

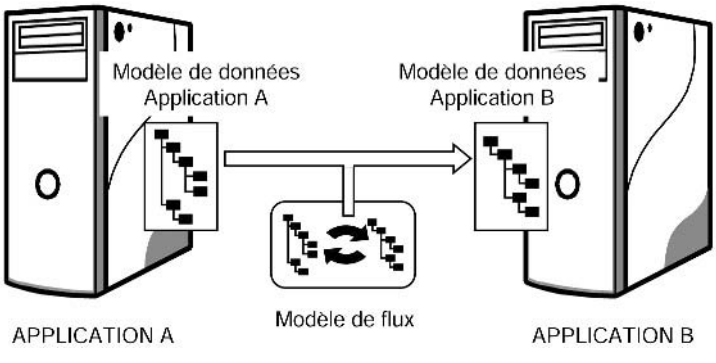

Figure 7.8 — Échanges directs de données entre applications

Les efforts de mise en œuvre d'une telle médiation sont importants et demandent une actualisation lourde et coûteuse pour chaque modification d'une des applications connectées. Ce ne sera pas le mode d'échange préféré autour des référentiels. On préférera de loin des outils « urbanisants » comme l'EAI ou l'ESB.

7.3.2 EAI (*Enterprise Application Integration*)

Beaucoup d'entreprises ont cherché à rationaliser leurs échanges à travers un bus de messages. Au final, cela s'est traduit par la mise en place des EAI (*Enterprise Application Integration*).

Les données sont transportées et transformées par l'EAI. L'objet spécifique des applications source est d'abord transformé en un format canonique (demi-flux entrant). Le format canonique est lui-même retransformé dans le format des objets

cibles (demi-flux sortant). Cet objet « pivot » ou « canonique » garantit l'indépendance des formats des données de chaque application. La figure 7.9 schématise ce type de communication.

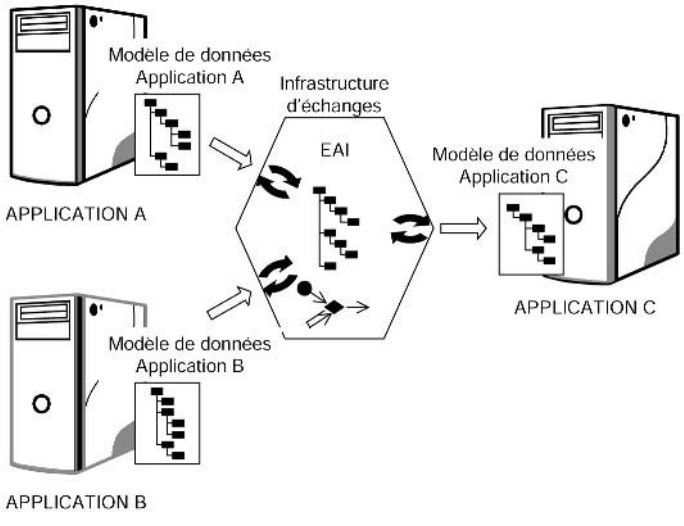

Figure 7.9 — Échanges via un EAI

La construction des échanges par demi-flux permet de simplifier les échanges. Ainsi, entre N applications il y a $(N-1)$! (factoriel $N-1$) flux possibles alors qu'avec un EAI il n'y a que N demi-flux[1]. La mise en œuvre réclame une attention particulière dans la construction du format pivot, mais génèrent des coûts de déploiement et de maintenance beaucoup moins importants que le « point à point » car seul le demi-flux concerné est impacté par l'évolution d'une application.

L'EAI est un support à l'urbanisation du SI. Il génère moins d'adhérence entre applications et plus d'agilité concernant l'évolution de l'architecture. Il offre aussi plus de réactivité en termes de mises à jour (mode message unitaire « au fil de l'eau »).

7.3.3 ETL (*Extraction Transformation Loading*)

Une nouvelle application (par exemple de décisionnel) ignore parfois où sont les sources de vérité. Par conséquent, on tente de récupérer les données de plusieurs applications et de les réconcilier au niveau du chargement par un ETL (*Extract Transform Load*). La figure 7.10 illustre ce type de situation. Dans ce cas de figure, on

1. Sur les EAI, voir F. Rivard et G. Abou Harb, *L'EAI au service de l'entreprise évolutive*, Maxima, 2003.

fait réaliser par l'ETL des fonctions de nettoyage et d'agrégation des données qui peuvent être délicates à mettre en œuvre et maintenir.

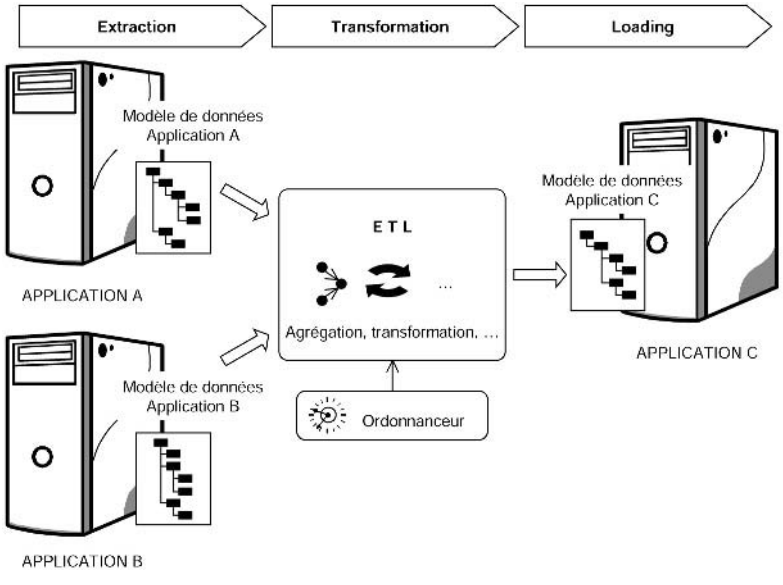

Figure 7.10 — Échanges via un ETL

En MDM, hormis les phases de chargement, on utilise l'ETL pour les architectures de consolidation ou pour alimenter des applications consommatrices à temporalité lente, sans obligation d'alimentation en continu.

7.3.4 Services et SOA

La SOA induit une structuration des données propres à leurs cas d'utilisation (services métier). Cette structuration implique une agrégation d'informations qui proviennent de diverses données à granularité fine afin de générer un service métier à forte granularité (*coarse grain*).

Cette encapsulation de services les uns au sein des autres, partant de services simples (service technique) pour aller vers une vision métier spécifique (service métier), peut être réalisée en :

- « codant en dur » cette encapsulation ;
- usant d'un outil SOA pour construire des services métier (un ESB ou un BPM par exemple).

À l'échelle du SI, il reste encore à identifier les sources de vérité de chacune de ces informations. Ce sont bien entendu les référentiels pour les données qui intéressent ici les auteurs. On remarque que plusieurs référentiels peuvent être mis à contribution pour générer un service métier, ce qui impose une gestion de la cohérence des référentiels entre eux (lien entre fournisseur et produit par exemple).

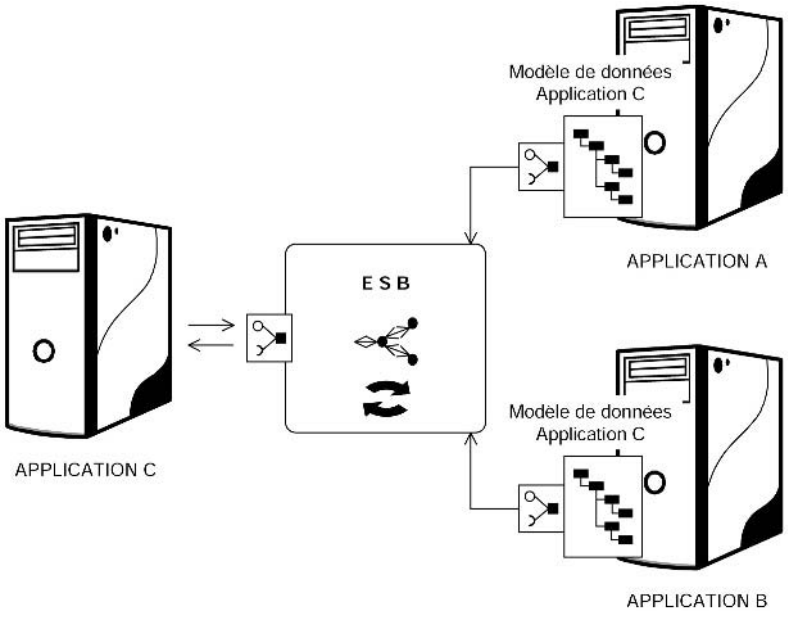

Figure 7.11 — Échanges via un service Web

À l'échelle des référentiels, la difficulté consiste à adapter la granularité des services à coder vis-à-vis de ceux à orchestrer. On se dote de services propres aux données constitutives du paradigme à couvrir, déclinées sous différentes fonctions (*add*, *modify*, *search*…) et on orchestre ces services entre eux dans un ESB afin d'offrir le plus d'agilité possible à l'architecture. Au même titre que l'EAI, l'ESB est donc un outil d'urbanisation du SI.

Par exemple, IBM MDM Server est aujourd'hui basé sur près de 680 services répartis en seize bibliothèques correspondant chacune à un des « objets constitutifs ». En jonction avec cet outil, on utilise de préférence un ESB pour créer des services métier adaptés au contexte d'utilisation de la solution.

Enfin il faudra dériver des cas d'utilisation les états supportés par la donnée, en lien avec son cycle de vie métier. Par exemple, le paradigme client peut recouvrir le prospect, le client en attente de devis, le client actif…

L'étude des cas d'utilisation, la définition des états supportés et l'effort de modélisation impliquent un effort d'analyse et de spécification plus important que les autres méthodes. Cet effort est compensé par la réutilisabilité.

Voir les annexes pour plus de détails sur la SOA et les liens entre données et SOA.

Voyons rapidement pourquoi les cabinets de prospectives secondés par les éditeurs claironnent que le MDM est une brique essentielle à la SOA.

Conséquences de la SOA sans MDM.

Les services sont rationalisés mais les données sont dispersées et, par conséquent, les services de manipulation de données métier sont plus complexes à développer. Cette complexité impose au SOA d'établir de nombreuses connexions vers les différentes sources de données (*mapping* inclus), demande une gestion constante des évolutions et conduit à une plus grande fragilité du système.

Conséquences du MDM sans SOA

L'intérêt métier du MDM reste entier. Cependant, au niveau du référentiel, l'intermédiation est plus lourde du fait des transformations entre les différents formats des consommateurs. L'effort d'urbanisation des échanges est moins normalisé qu'en SOA.

Remarquons que l'argument ne tient que pour les solutions SOA où on peut imposer un modèle normalisé de la donnée qui, en toute logique, est celui du référentiel. Dans une architecture SOA incluant des progiciels, seuls les processus transverses bénéficient du modèle normalisé. Nous serons encore longtemps obligés de réaliser des *mappings* de données.

Bénéfices d'une architecture SOA et MDM

La mise à disposition des données au travers de services implique la rationalisation des services et des données manipulées par les services. Le MDM permet cette rationalisation et devient un accélérateur dans la mise en place du SOA.

7.3.5 MDM et échanges, apports et nécessités

Si on développe la démarche MDM, celle-ci concourt à la rationalisation des échanges et s'impose comme point focal dans le SI. En amont, les applications source l'alimentent en direct ou une interface permet la saisie. En aval, les applications consommatrices s'y rattachent pour déterminer la source de leurs données.

Concernant les échanges, on passe historiquement d'une vision point à point « spaghetti » (1) à une vision urbanisée (2), pour enfin arriver à une vision gouvernée des données (3) (unique point de *sourcing*, qualité, métadonnées, maîtrise de la propriété…) (*cf.* figure 7.12).

Dernier avantage, la mise en place du MDM permet une rupture de temporalité par rapport à la vision dynamique induite par les échanges. Ceux-ci étant tributaires des processus sources ou ordonnancés, la communication entre les applications requiert, soit de se plier au mode d'envoi, soit de multiplier les moyens de communication afin de répondre aux besoins des consommateurs.

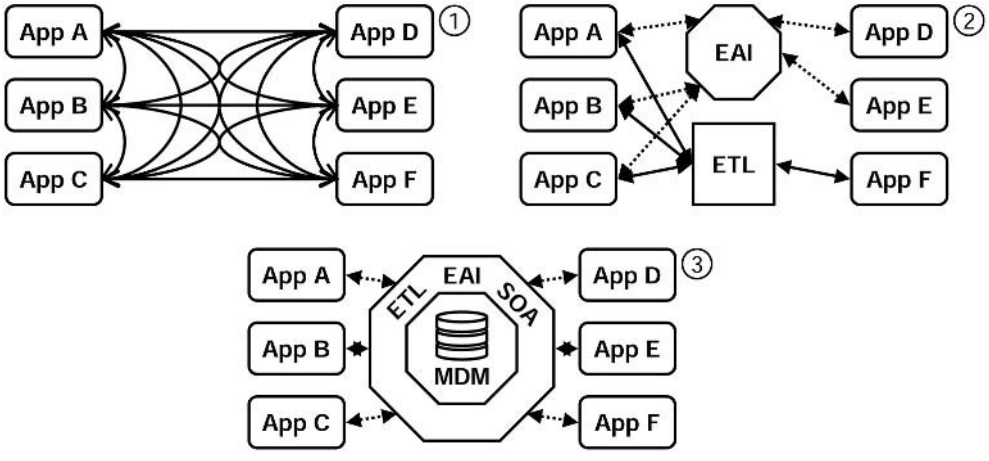

Figure 7.12 — Évolution de la prise en compte des données

La mise en œuvre d'un MDM apporte, entre les techniques d'intermédiation, le même type de simplification que l'EAI entre les flux. Avec un MDM opérant au cœur d'un SI, non seulement chaque échange est régi par demi-flux, mais chaque demi-flux supporte un unique vecteur d'intermédiation, celui le plus adapté à l'application connectée.

Nous avons vu précédemment que deux applications source offraient chacune une extraction ETL, un mode message EAI et un service Web. Pour répondre aux besoins de trois applications utilisatrices sur un même paradigme, on comptait donc six interfaces. Le MDM induit seulement cinq interfaces. Ici, ce n'est pas tellement la réduction du nombre d'interfaces qui nous intéresse, mais surtout que chacune soit parfaitement adaptée à l'application ou à l'utilisation de la donnée.

Dans notre exemple, les applications source sont connectées l'une en EAI, l'autre en ESB et les applications consommatrices consomment l'une par ETL, l'autre par EAI et la dernière, par requête sur service Web. Le MDM assure la continuité transactionnelle pour l'application abonnée par EAI. Pour les deux autres applications consommatrices, le MDM sert de barrage et ne libère de réponse, soit en masse (ETL), soit unitairement (services Web) qu'après sollicitation.

Si cette possibilité ne vous convainc pas dans le périmètre d'un SI existant, imaginez la simplification que cela induit lors de l'ajout d'une application nouvelle. L'introduction d'une nouvelle application ne nécessite alors que la mise en place d'une unique interface par donnée de référence.

Outre les fonctions de normalisation, de gestion des droits, d'historisation, de maîtrise des métadonnées ou de gestion de version induites par le MDM au bénéfice

des échanges, ceux-ci bénéficient d'une décorrélation aussi bien logique que technique grâce à cette brique.

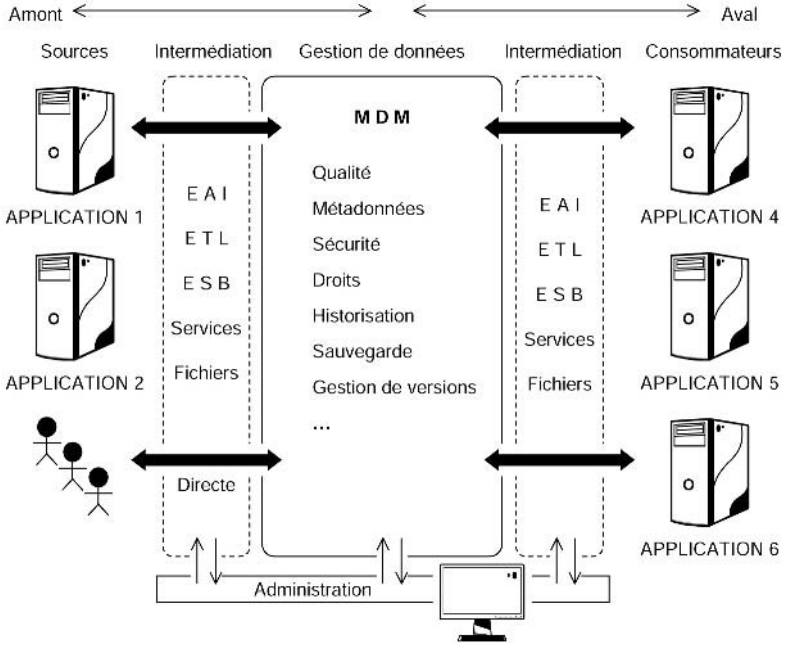

Figure 7.13 — La MDM, une brique de décorrélation logique et technique

7.4 POUR UNE INSERTION PROGRESSIVE DU MDM DANS LES ÉCHANGES DU SI

Insertion du MDM dans le SI

On peut considérer qu'une donnée « voyage » dans le système d'information entre le moment où elle entre dans le SI et son ultime point de consommation.

Dans un SI sans solution de gestion des données de référence, une donnée de référence passe d'application en application au sein du SI transactionnel et jusqu'au SI décisionnel.

Par exemple, dans la grande distribution, une donnée « produit » provient en partie d'un fournisseur et en partie de divers services de l'entreprise (et donc de diverses applications dans le SI) tels que la logistique, les achats ou le marketing. Cette donnée peut transiter par diverses applications pour finir avec l'application « ligne de caisse » et le SI décisionnel comme consommateurs ultimes.

Dresser un tel état des lieux, une cartographie de l'existant, permet de mieux identifier où devra se situer la solution référentielle au sein de la chaîne de l'information (figure 7.14).

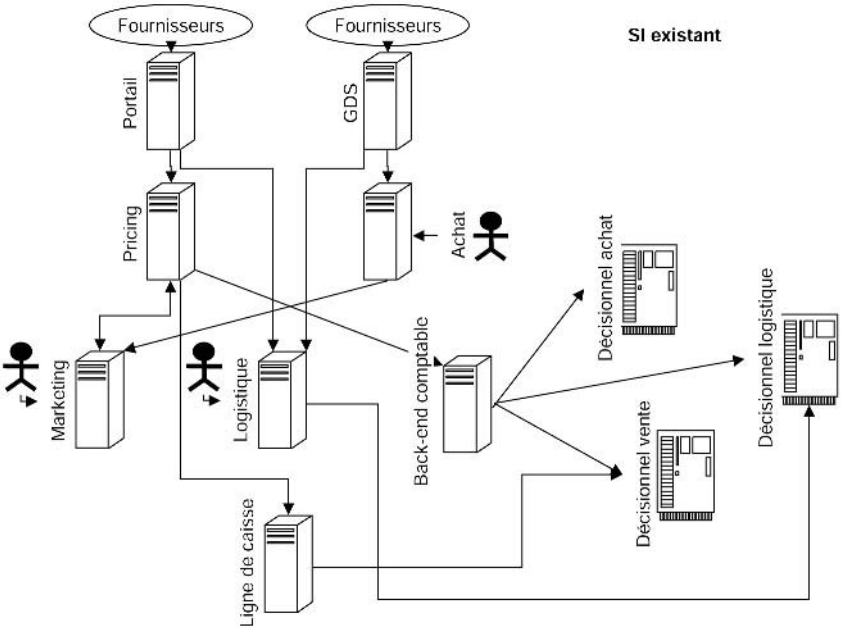

Figure 7.14 – Exemple de SI existant de la grande distribution

Cette cartographie de l'existant permet aussi de prévoir un plan de migration afin de rationaliser la « cascade » d'applications. On cherche notamment à mettre toutes les applications source ou consommatrices à un seul niveau du référentiel (en étoile autour du référentiel). Cette « migration » est établie en fonction de la criticité des applications liées, du portefeuille d'applications et de leur calendrier de remplacement au sein du SI…

Dans le SI d'une grande entreprise, cette évolution va potentiellement s'étaler dans le temps, sur une longue période, en passant par une (ou plusieurs) étape(s) intermédiaire(s) (figure 7.15) avant d'atteindre une cible (figure 7.16).

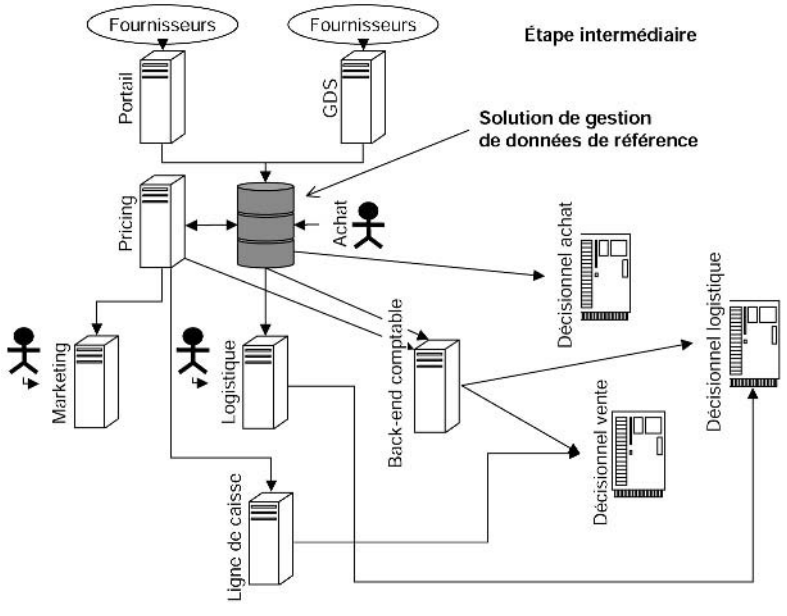

Figure 7.15 – Évolution du SI par rapport à la solution référentielle (étape intermédiaire)

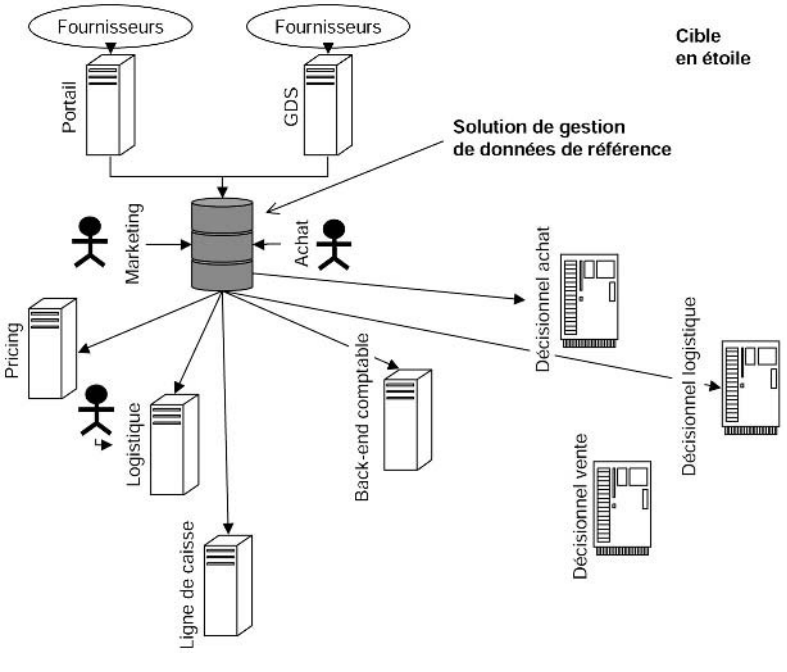

Figure 7.16 — Évolution du SI par rapport à la solution référentielle (étape finale)

Pour préparer votre « plan de transformation référentiel », identifiez les processus amont (ceux qui participent de la constitution de la donnée) et les processus aval. Identifiez les applications qui supportent ces processus. À partir de cet inventaire, dressez une carte de la situation actuelle et, au travers du portefeuille projet, dressez la carte de votre cible.

Pondérez chaque processus en fonction de sa criticité pour l'entreprise. Votre analyse combinera ainsi pérennité des applications et criticité afin d'établir le raccordement de la solution de gestion de données de référence à votre SI.

En toute logique, la solution référentielle doit se situer au plus haut de la chaîne de consommation de la donnée, afin de fournir une information valide à l'ensemble du SI, puis devenir le centre d'une couronne d'alimentation. L'architecture de centralisation est, suivant cette règle, celle qui permet une parfaite maîtrise de la donnée. Ensuite, les capacités de gestion des données et de pilotage décroissent suivant l'ordre suivant des architectures : coopération puis consolidation et enfin répertoire virtuel.

Notion de propriétaire

La cartographie des applications source, alliée à la nature du référentiel, induit des notions de propriété (voir plus en détail ces notions dans la troisième partie). Il faut distinguer les propriétaires du modèle et des instances.

La notion abordée ici porte sur le propriétaire du modèle qui est induit par la nature du référentiel et sur les propriétaires d'instances qui sont induits par la cartographie.

Propriétaire d'instances

En aval, le référentiel est propriétaire des instances d'objet pour chacun des attributs du paradigme qu'il détient.

En amont, une application supporte un ou plusieurs rôles utilisateurs intervenant sur l'application. Dans la mesure où on peut identifier les attributs détenus par l'application, on peut donc dresser les corrélations entre rôles et attributs.

Propriétaire de modèles

Au sein d'un référentiel globalisant, plusieurs métiers sont responsables de la définition du modèle de donnée puis de la complétion des données. **Plusieurs propriétaires** sont chacun responsables de leur périmètre de données. Un responsable pour la cohérence de l'ensemble doit cependant valider l'évolution du modèle.

Pour les référentiels synthétiques, la définition du modèle s'apparente plus à la négociation d'un format pivot (**un seul propriétaire**).

Ce travail induit une documentation précise du modèle. Cette documentation commence dès la définition sémantique de chaque concept (glossaire) puis de chaque attribut (dictionnaire). Ces définitions doivent être liées aux applications de métadonnées.

La copropriété d'un objet métier signifie la répartition stricte de chaque attribut. Il n'y a pas de multipropriété pour un unique attribut au niveau du modèle.

En résumé

Les solutions de référentiel reposent d'abord sur les applications de MDM. Mais, répondre aux besoins métier liés à la solution MDM, intégrer cette solution dans le SI et en permettre le pilotage peut demander l'utilisation de briques transverses, de modules spécifiques ajoutés ou de briques complémentaires progiciel ou *middleware*. Il faut porter une attention particulière aux applications d'intermédiation.

Cette intégration ne se fait pas en mode « big bang » mais par étapes à maîtriser en repérant les processus amont et aval et en élaborant un plan de transformation du SI. Il s'agit de faire du référentiel la pièce centrale d'une architecture en étoile.

8

Guide de choix des architectures et solutions

Objectif

Comment choisir l'architecture (centralisation, consolidation, coopération) adaptée à la gestion d'un type de donnée de référence ?

Quelle solution (MDM, DQM, progiciel, développement spécifique) adopter ?

Quand privilégier en particulier une solution MDM ?

Telles sont les principales questions auxquelles nous répondrons dans ce chapitre. Nous évoquerons aussi les bonnes pratiques métier, urbanisme et architecture.

8.1 CHOIX D'ARCHITECTURE

Nous avons abordé plusieurs critères de choix d'architecture liés à des objectifs et des besoins fonctionnels lors de leur présentation dans le chapitre 4. Nous les complétons ici avec quelques critères plus techniques qui sont détaillés dans le tableau 8.1.

Rappelons toutefois qu'au sein d'un SI il peut y avoir plusieurs référentiels et que le choix d'une architecture s'effectue par donnée de référence (une même solution abritant plus d'un référentiel peut donc implémenter plusieurs architectures).

Tableau 8.1 — Guide complémentaire de choix d'architecture

Critères	Commentaires
Couplage lâche : les applications consommant la donnée de référence sont-elles découplées des applications fournissant la donnée de référence (désynchronisation) ?	Par défaut, toutes les architectures sont compatibles, avec une facilité accrue pour les architectures de consolidation qui jouent bien le rôle de « concentrateur de la donnée », sans se lier avec une application en particulier.
Distribution : les données de référence sont-elles distribuées vers de nombreuses applications destinataires ?	Comme son nom l'indique, une architecture de consolidation fédère plus les données qu'elle ne les distribue. L'architecture de centralisation est celle qui convient le mieux pour distribuer.
Recherche et analyse : est-il demandé d'avoir des services de type recherche, analyse et pilotage ?	Toutes les architectures sont aptes à ce type de critère.
Préservation de l'existant, accompagnement : souhaite-t-on éviter une conduite de changement trop forte dans un premier temps (exemple : préservation du patrimoine applicatif) ?	L'architecture de coopération permet de fournir des écrans et des *workflows* propres aux applications existantes.
Référentiel existant : existe-t-il un référentiel progiciel de fait (par exemple ERP) ? Existe-t-il une base de données de concentration ?	Ce référentiel progiciel existant est par nature une architecture de centralisation. Une base de données de concentration est généralement un premier essai en développement spécifique pour une architecture de consolidation
Processus d'acquisition : l'objectif poursuivi est-il la simplification des processus d'acquisition ?	Les architectures de centralisation permettent de construire et de valider les processus complexes d'acquisition de la donnée. Une architecture de coopération peut répartir ce processus sur plusieurs applications source, coordonnées du point de vue événementiel par la solution de gestion des données de référence.
Qualité : est-ce un problème de qualité de la donnée ?	Les architectures de consolidation et de coopération doivent proposer des mécanismes de gestion de la qualité de la donnée, les sources n'étant pas maîtrisées par la solution. L'architecture de centralisation, par essence, travaille au travers des processus sur une source d'acquisition de qualité : le référentiel lui-même. Les processus peuvent utiliser un outillage DQM, pour répondre à un besoin de traitement complexe (standardisation, déduplication).

Voir aussi la figure 4.13 pour un rappel illustré des principaux critères de choix d'architecture. La figure 8.1 synthétise et simplifie ces critères.

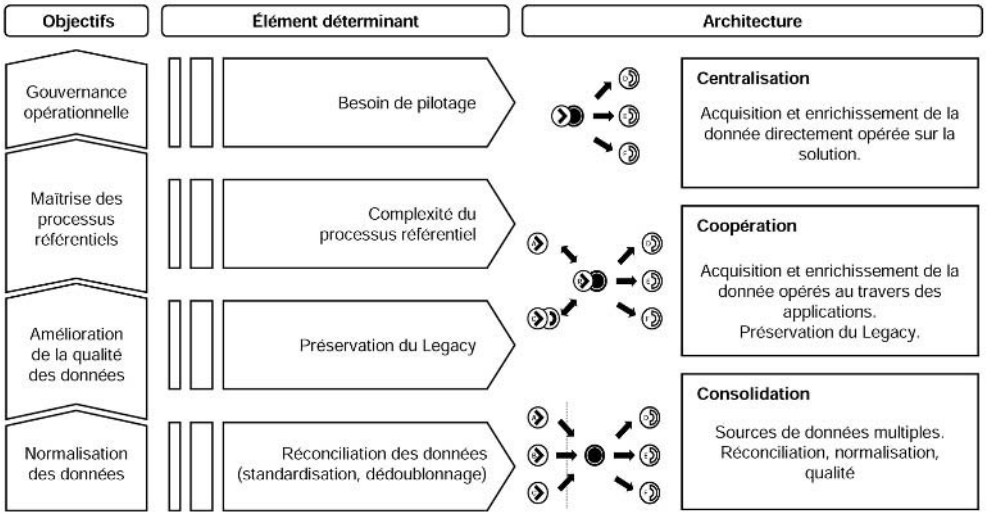

Figure 8.1 — Rappel des principaux critères de choix d'architectures

8.2 CHOIX DE SOLUTIONS

Nous avons évoqué plusieurs types de solutions et leurs besoins associés au chapitre 5. Nous nous focalisons ici sur la gestion des données de référence afin de déterminer si elle doit être effectuée par :

- un développement spécifique ;
- un progiciel métier (ERP, CRM) ;
- une solution de MDM (que l'on peut assimiler à un progiciel spécifique pour la gestion des données de référence).

Le tableau 8.2 et la figure 8.2 donnent quelques éléments de choix.

Notons à nouveau que les sous catégories d'outils MDM peuvent répondre différemment aux critères exprimés ici. Notre analyse ne prend pas en compte cette segmentation. De plus, une solution MDM couvre, à notre sens, les applications référentielles MDM ainsi que le DQM ; nous ne faisons donc pas la distinction.

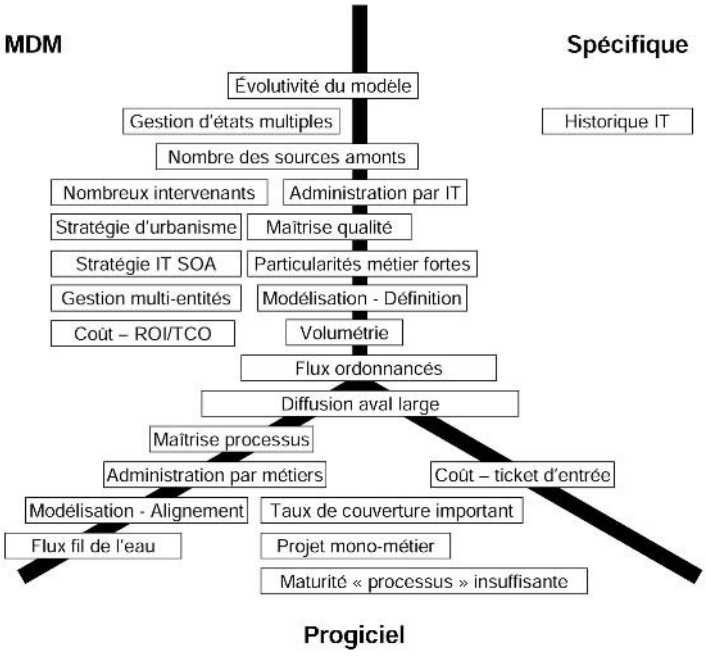

Figure 8.2 – Critères différenciateurs entre MDM, développements spécifiques et progiciels

Tableau 8.2 — Guide de choix de solutions

Critères	Commentaire
Modélisation L'entreprise travaille-t-elle plus par définition de ses objets ou par alignement ? Existe-t-il déjà un modèle d'objet métier pour le paradigme concerné (pivot, développements spécifiques…) ?	Dans le cas de la définition ou de l'intégration d'un modèle existant, le MDM et le développement spécifique sont seuls à offrir suffisamment de flexibilité. Le spécifique demande une modélisation physique, contrainte par le relationnel. Le MDM se contente généralement d'une modélisation logique ou offre une modélisation flexible (XML). Dans le cas de l'alignement, MDM et progiciel répondent au besoin. Les solutions progicielles viennent toutes avec un modèle et une méthode d'alignement. Les MDM arrive généralement avec des modèles pré-intégrés servant d'accélérateur.
Évolutivité du modèle Le périmètre du paradigme concerné sera-t-il mis en œuvre par itération ? L'objet est-il appelé à évoluer fréquemment ou reste-t-il stable dans le temps ?	Dans le cas d'une évolutivité forte ou d'une mise en œuvre par itération, le MDM et le développement spécifique sont préférables. Les contraintes de modélisation du spécifique limiteront cependant son évolutivité à moyenne échéance. Les solutions progicielles contraignent des périmètres de mise en œuvre (au niveau modèle, droits, règles, cohérence).

Critères	Commentaire
Gestion d'états multiples Les cas d'utilisation, au long du cycle de vie métier de la donnée, sont-ils nombreux, multipliant ainsi les états que la donnée doit supporter ?	Seul le MDM permet la gestion de multiples états de la donnée avec les combinaisons de règles et de droits propres à chaque état. Le développement spécifique est disqualifié.
Particularités métier fortes L'objet à supporter est-il très spécifique, propre à un métier peu répandu ? (cas des administrations)	Le développement spécifique semble être le plus indiqué, mais en apparence seulement. Le MDM offre en mode projet la flexibilité de la modélisation et une plus grande maîtrise des métadonnées (modèle et sémantique) et permet à terme plus de flexibilité pour l'évolution.
Volumétrie Quel est le nombre d'occurrences de l'objet ou d'ID ? Quel coefficient multiplicateur doit être appliqué en raison de l'historisation ?	La volumétrie en nombre d'objets peut aller de quelques centaines d'occurrences à plusieurs centaines de millions d'enregistrements. Sur les fortes volumétries (plusieurs millions), les progiciels seront distancés en capacité de gestion, si ce n'est en nombre d'enregistrements. Les solutions MDM ne sont pas toutes équivalentes face à la volumétrie et une étude particulière pour telle ou telle solution de tel ou tel éditeur devra donc être menée, d'autant plus si les capacités fonctionnelles attendues incluent des recherches complexes. Certaines solutions supportent cependant de très fortes volumétries (dizaines de millions). Le développement spécifique peut tirer son épingle du jeu sur les très grandes volumétries (plusieurs dizaines voire centaines de millions), couplées à des attentes particulières (modèle ou fonctions).
Taux de couverture des progiciels La couverture offerte par le modèle de donnée du progiciel en pourcentage du nombre d'attributs, ainsi qu'en criticité des attributs détenus, est-elle importante ? L'écart entre le modèle détenu et la cible, ainsi que les fonctions ou règles à développer n'impliquent-ils pas un éloignement trop important du standard du progiciel ?	Le taux de couverture fonctionnel et celui du modèle doivent être maximaux pour être compatibles avec la mise en œuvre d'un progiciel. Les écarts sont gênants pour le support et les montées de version.
Gestion des droits – nombre intervenants Les intervenants humains ou machines sont-ils nombreux, multipliant les profils ? Les droits doivent-ils être gérés avec précision, sur des périmètres très segmentés ou en multipropriété ?	Plus les profils et les besoins de maîtrise des droits sur la donnée sont importants, plus le MDM permet une gestion fine (au niveau attributs, fonctions, par rapport aux valeurs) et aisée (profils, groupe de profils, duplication de profils pour création…).

Critères	Commentaire
Applications connectées – amont	Plus le nombre d'applications source est important, plus le MDM est préférable. Que ce soit en coopération pour la maîtrise transactionnelle ou en consolidation pour les capacités de déduplication et de *key mapping*.
Applications connectées – aval	Plus le nombre d'applications cibles est important, plus le MDM offre de flexibilité en termes de mode de connexion, de décorrélation avec les sources, de gestion des processus de synchronisation.
Flux « fil de l'eau »	Les solutions progiciel ou MDM supportent mieux le « fil de l'eau ». Les progiciels induisent souvent des traitements « fil de l'eau » ou synchrones. Le MDM reste favori face au progiciel pour les fonctions supplémentaires qu'il apporte (gestion événementielle).
Flux ordonnancés – en masse	Les flux ordonnancés et/ou en masse ne sont pas tellement différenciateurs. Léger avantage cependant au MDM qui offre toujours sa flexibilité et intègre la gestion événementielle. Mode classique, par extraction ou duplication sur le spécifique. Mode classique sur les progiciels.
Fonctions qualité La solution doit-elle couvrir des besoins de standardisation complexe (adresse, acronyme, abréviation) ou de déduplication ?	Seule la solution MDM offre un support complet à la qualité (en lien avec le DQM). Le développement spécifique en fin de chaîne, de l'information s'il est couplé à un DQM, peut offrir un certain niveau de qualité en redressement (cas du BI standard). Le progiciel n'offre pas de fonction qualité.
Historique IT Les compétences et les habitudes de la DSI sont-elles plus orientées spécifique ou progiciel ?	Les entreprises ayant une forte culture du développement spécifique et/ou ayant de nombreuses applications et compétences internes pour gérer le spécifiques passeront plus difficilement au progiciel ou au MDM. Si le pas est franchi, il se fera au bénéfice du MDM plutôt que du progiciel. Les progiciels sont généralement supportés par les métiers.
Administration – métiers ou IT L'administration de la solution de gestion des données de référence est-elle réalisée par les métiers ou l'IT ?	Si les équipes d'administration pressenties sont IT alors le MDM et le développement spécifique sont plus adaptés. Le MDM reste préférable car il offre des fonctions propres à l'administration. Si les équipes d'administrations sont métiers, alors elles ont tendance à préférer les progiciels. Mais ce point sera contrebalancé au profit du MDM si les intervenants métier sont nombreux.

Critères	Commentaire
Stratégie DSI – urbanisme – SOA L'entreprise est-elle dotée d'une cellule d'urbanisme ? Les préceptes de la discipline influent-ils sur le SI ? La démarche référentielle répond-elle à une vision d'ensemble, généralisée ? La DSI est-elle engagée dans une stratégie IT orientée SOA ? ou du moins profite-t-elle des outils urbanisants (par exemple, EAI) ?	Le MDM est une réponse logique d'urbaniste, d'autant plus adaptée que la démarche est généralisée (certains outils MDM peuvent couvrir de multiples paradigmes). Idem dans le cadre d'une stratégie SOA, le MDM est considéré comme une brique constitutive de cette stratégie.
Besoin de gestion multi-entité, multimétier ou mono-métier, mono-entité La solution doit-elle couvrir plusieurs métiers, plusieurs entités organisationnelles ?	Plus la solution est répartie entre entités et sur plusieurs métiers, moins le progiciel est adapté. MDM et spécifique restent utilisables, avec un net avantage pour le MDM du fait de ses capacités de gestion (droits) et de contextualisation.
Coût – ticket d'entrée L'entreprise a-t-elle estimé les coûts d'un projet référentiel à leurs justes valeurs ou a-t-elle tendance à considérer que cela revient à mettre une simple base dans une partie du SI ?	Le spécifique semble toujours offrir un coût facial moindre. Cela reste vrai sur de petits projets. La mise en place d'un progiciel pour le métier (un CRM par exemple) peut aussi être ressentie comme l'opportunité d'utiliser celui-ci sans générer de coûts supplémentaires. Seule une approche tactique permet la diminution de ce ticket d'entrée pour le MDM, sans l'annuler.
Coût - ROI/TCO	Sur des projets de plus grande ampleur, l'analyse ROI/TCO profitera au MDM grâce à la diminution des points de gestion multiples dans le SI et l'amélioration des processus métier consommateurs (plus étendue qu'avec un simple progiciel qui n'améliore que ses propres processus).
Immaturité « processus » de l'entreprise L'entreprise a-t-elle déjà documenté et outillé ses grands processus (par exemple, « *Order to Cash* ») ? Ces processus ont-ils déjà été améliorés ?	Dans une entreprise qui n'en est qu'au début de l'outillage de ses processus, il y a beaucoup plus de valeur à générer par la mise en place des progiciels qui vont les structurer que par l'outillage de la donnée. On utilise dans une telle phase les progiciels comme référentiel. Seule exception, quand le processus métier est un processus de gestion de données (par exemple référencement produit dans la grande distribution, ou gestion des catalogues pour les services achats), on outille préférentiellement avec une solution MDM adaptée.
Maîtrise des processus La solution de gestion de données de référence outille-t-elle les processus d'acquisition de la donnée ? Ces processus sont-ils complexes, font-ils participer de multiples profils intervenants ?	Le développement spécifique est disqualifié. Le progiciel couvre les seuls processus prévus à son périmètre. Avantage donc au MDM qui généralement favorise la modélisation des processus d'acquisition.

La figure 8.3 résume les conclusions pour le choix de solutions.

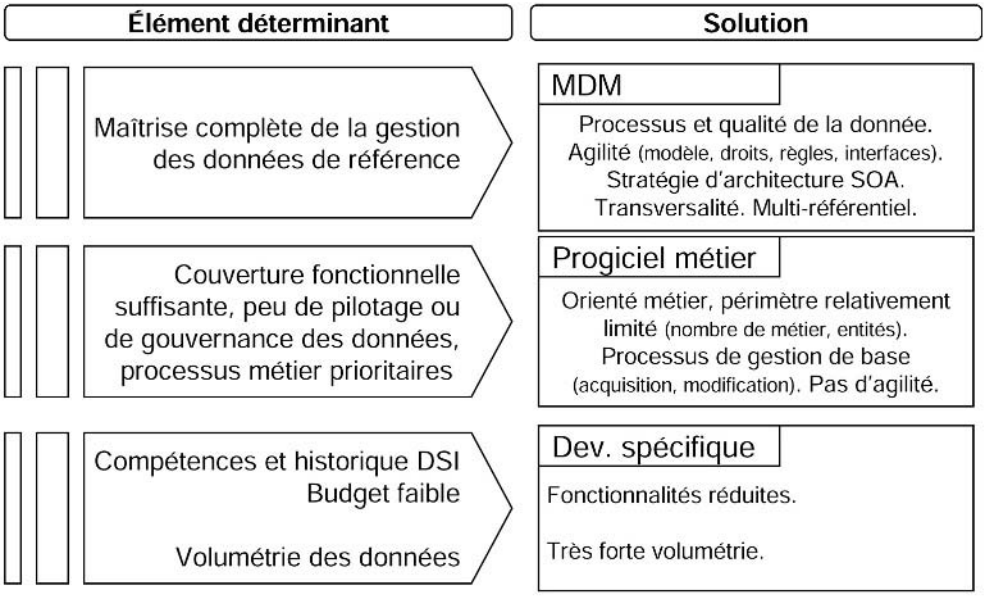

Figure 8.3 – Principaux critères et choix de solutions

8.3 MODE D'IMPLÉMENTATION ET ÉLIGIBILITÉ DES SOLUTIONS DE MDM

Nous avons vu dans le chapitre 4 les modes d'implémentation dérivés des architectures types.

Suite aux critères de choix que nous venons d'étudier, la figure 8.4 résume l'éligibilité des solutions en fonction des modes d'implémentation.

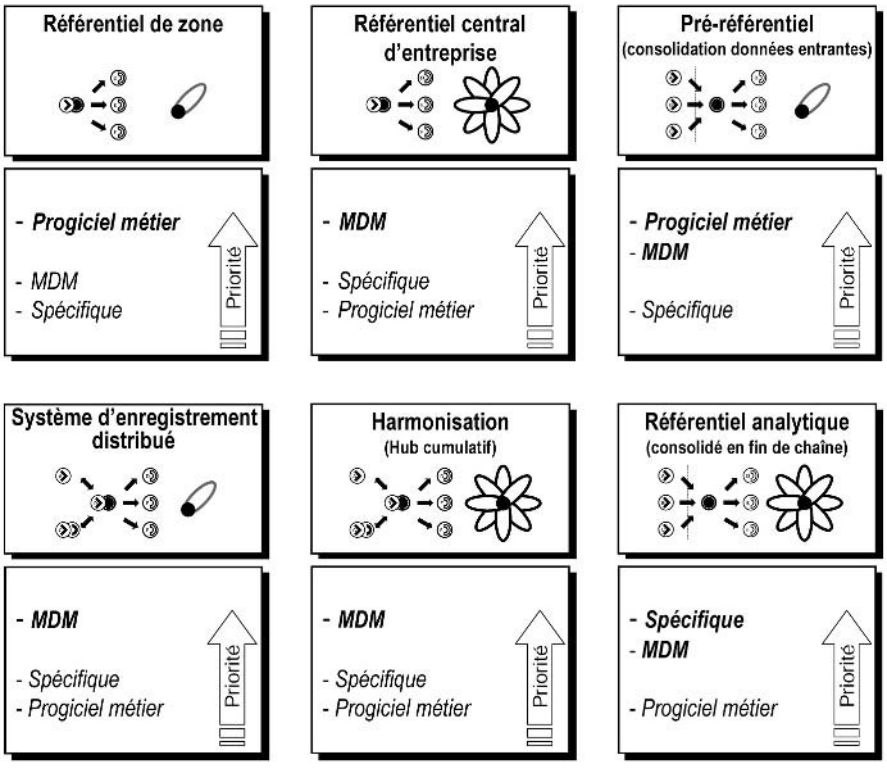

Figure 8.4 — Éligibilité des types de solution aux modes d'implémentation

On voit que les solutions de MDM sont à envisager prioritairement pour « Référentiel central d'entreprise », « Harmonisation », et « Système d'enregistrement distribué ».

La mise en œuvre du MDM est moins prioritaire pour le « Pré-référentiel » (cela dépend du type de données) et pour le « Référentiel analytique » (un logiciel spécifique allié à du DQM peut suffire).

La figure 8.5 indique le choix du mode d'intermédiation en fonction du mode d'implémentation : mode fichier (appelé « *batch* » sur la figure, total ou delta) ou message (« fil de l'eau » ou synchrone). Nous avons simplifié les modes de connexion sans entrer dans le détail des protocoles.

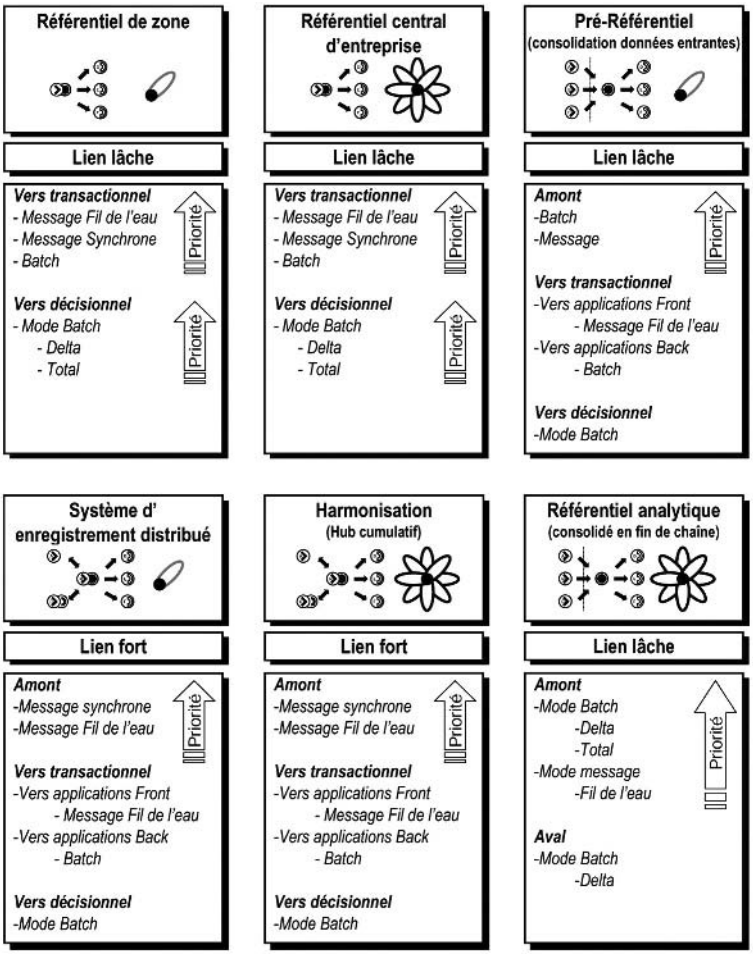

Figure 8.5 – Préférence de choix des modes d'intermédiation

8.4 SOLUTIONS MISES EN PLACE PAR QUELQUES ENTREPRISES

8.4.1 Un grand distributeur

Le tableau 8.3 indique les principales caractéristiques de la mise en œuvre des données de référence du domaine Marchandise, dans une entreprise de la grande distribution. Les données concernées sont essentiellement de type Fournisseurs et Produits.

Tableau 8.3 – Caractéristiques de la mise en œuvre de la gestion des données de référence de la société X (grande distribution)

Besoins identifiés, solutions mise en place	**Besoins principaux identifiés** – Diminuer les coûts et rendre le SI plus évolutif. – Éviter les saisies multiples. – Assurer la qualité des données. – Diminuer le temps de mise en rayon d'un nouveau produit. – Assurer une maîtrise complète du processus d'acquisition de la donnée et de l'ensemble des processus connexes, en garantissant la qualité ainsi que les impératifs IT et la diffusion. – Un pilotage de l'ensemble est permis par la mise en place et la surveillance d'indicateurs de fonctionnement. Le besoin étant très proche du métier et s'inscrivant dans une approche de refonte et de maîtrise du processus référentiel, les applications nécessaires dépassent les simples applications de référentiels. **Solutions mises en place** : Elles sont distinctes en fonction de leur place dans la chaîne référentielle (voir figure 8.6) : – Pré-référentiel : logiciel MDM + portails sélectionnés après étude de marché et choix de solution sur POC (*Proof Of Concept*). Le pré-référentiel supporte plusieurs modes d'acquisition (portail en saisie directe, flux spécifiques, synchronisation avec *market place*). – Enrichissement : *workflow* + EAI. – Référentiels locaux : différentes solutions en fonction des pays – MDM pour les grands pays ou progiciel métier pour les petits pays.
Principaux bénéfices et facteurs de succès	**Mise en place d'un centre de compétence** : le centre de compétence est la clef de voûte de la démarche. Il allie compétences métier et compétences techniques. ***Re-engineering* des processus métier :** la nature de la solution référentielle a nécessité une importante phase de refonte des processus métier (processus de référencement des fournisseurs, processus achat, processus de référencement des articles). Les processus ont été entièrement documentés, les rôles et acteurs décris et chaque notion métier explicitée au sein d'un glossaire. **Gestion du changement :** la refonte des processus métier, et par voie de conséquence des interfaces de gestion, nécessite un accompagnement de l'ensemble des populations d'utilisateurs.
Écueils à éviter	**L'inertie et la résistance aux changements des pays sont le principal frein à la mise en œuvre lors du déploiement.** L'organisation, l'utilisation des budgets, l'incitation par des projets pilotes et une communication interne intense sont autant d'exemples des actions entreprises afin de briser cette résistance. **Sous-estimer les difficultés techniques**, notamment avec une architecture complexe et des produits manquant d'interopérabilité peut rallonger les délais du projet.

La figure 8.6 illustre les solutions retenues.

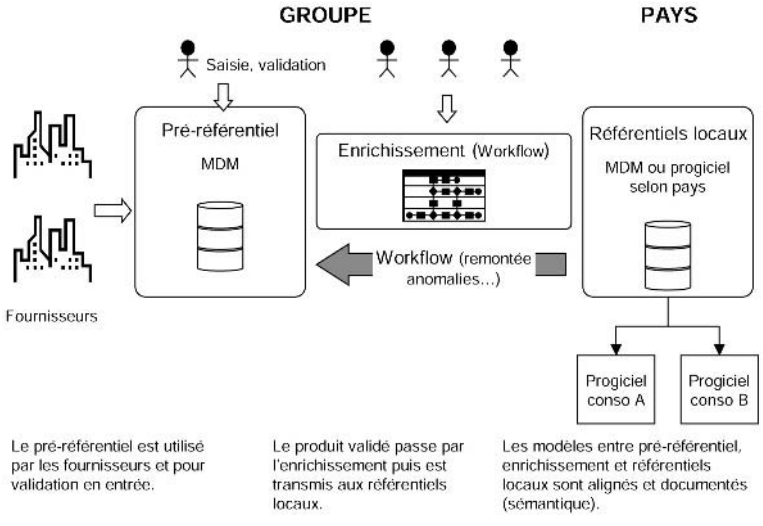

Figure 8.6 — Solutions retenues par la société X (distributeur)

8.4.2 Un producteur

Le tableau 8.4 indique les principales caractéristiques de la mise en œuvre d'un référentiel client chez un producteur industriel.

Tableau 8.4 — Caractéristiques de la mise en œuvre de la gestion des données de référence de la société Y (producteur industriel)

Besoins identifiés, solution mise en place	**Besoins identifiés** : les besoins ont été identifiés (maîtrise de la qualité des données, mises à jour et diffusion plus rapides...) mais la mise en place, fortement poussée par la DSI, tient de la démonstration dans le cadre d'une stratégie SOA et d'une modernisation complète du SI. **Solutions mise en place** : Produit MDM pour le référentiel + produit DQM pour la qualité des données (les fonctions de qualité intégrées au MDM ne suffisant pas dans ce cas). Les autres référentiels ne sont pas encore outillés, un choix de solution sera effectué en réponse aux besoins de chacun. Cependant, la société Y considère qu'une même application de type MDM doit pouvoir couvrir plusieurs besoins. Le progiciel MDM retenu doit répondre aux référentiels orientés personnes ou organisation (clients, fournisseurs, employés, par exemple). De plus, la société Y analyse les solutions complémentaires (PLM pour le référentiel des produits semi-finis par exemple).
Principaux bénéfices et facteurs de succès	**Principaux bénéfices** : la normalisation induite par la démarche référentielle induit **une simplification et une mutualisation des ressources projets**. C'est néanmoins difficile à chiffrer. **Facteurs de succès :** les projets avancent à pas mesurés mais s'inscrivent dans une vision globale de gouvernance. Le premier projet est celui qui rassemble l'avantage d'être le plus visible/sensible aux métiers tout en permettant une mise en œuvre raisonnée de la solution (d'abord le référentiel, puis le DQM, d'abord en consolidation derrière les référentiels existants puis vers une architecture de coopération). Une organisation directement responsable de la donnée est en charge des règles mais aussi de leur bonne application. Implication des métiers depuis le niveau stratégique puis à chaque niveau des projets. L'éditeur est impliqué en expertise sur les produits.
Écueils à éviter	**Le premier projet de l'initiative MDM était trop ambitieux**, en faisant abstraction d'un existant important et structurant (en termes de processus et de liaison avec les systèmes historiques). **L'implication des métiers dépend de la valeur que les projets référentiels peuvent leur apporter. Cependant, cette valeur est difficilement démontrable** car les projets référentiels sont essentiellement des projets de support aux processus et aux applications métier mais n'apparaissent pas directement aux métiers. La valeur générée est donc indirecte.

8.4.3 Un fournisseur

Le tableau 8.5 indique les principales caractéristiques de la mise en œuvre du référentiel achat (fournisseurs, articles).

Tableau 8.5 – Caractéristiques de la mise en œuvre de la gestion des données de référence de la société Z

Besoins identifiés, solution mise en place	**Besoins identifiés** : maîtriser les référentiels et apporter de la souplesse dans leur gestion. Visibilité unifiée des fournisseurs pour la Direction des achats, flexibilité du SI, partager une infrastructure de gestion commune, assurer la qualité des données. **Solutions mise en place** : progiciel MDM sélectionné après étude du marché. Retenu surtout pour les modèles existants par rapport au métier. Architecture centralisée.
Principaux bénéfices et facteurs de succès	**Principaux bénéfices** : gestion maîtrisée des référentiels achats. **Facteurs de succès** : **– La démarche « démarrer petit pour voir grand »** permet de se familiariser tant sur les concepts que sur la définition des rôles pour le MDM mais également sur la technologie du progiciel retenu et la gestion de la qualité de la donnée. – Implication de l'éditeur sur le projet qui permet d'accompagner la société Z sur les choix de briques de solution. – Implication nécessaire de la Direction Achats. – Accompagnement indispensable par rapport à la solution.
Écueils à éviter	**L'initiative MDM manque de sponsoring des métiers** ce qui limite la portée en termes de communication. L'apport du pilote réalisé a été affaibli car le périmètre était trop restreint tant sur le plan fonctionnel que sur les données de référence (nombre d'occurrences, types de données, nombre de rôles…) ce qui ne permet pas d'extrapoler sur un référentiel plus ambitieux. **Le MDM est une composante d'infrastructure qui doit être mutualisée pour maximiser le ROI et assurer des gains de productivité et d'industrialisation.**

8.4.4 Conclusion

Comme nous l'avons évoqué dans la première partie de l'ouvrage, les constats sur la gestion des données font clairement apparaître le besoin d'une fonction « référentiel des données de référence ». Néanmoins, ces référentiels s'appuient bien souvent sur des développements spécifiques réalisés au cours du temps ou des progiciels métier.

Certaines entités lancent des études sur la possibilité de mieux gérer les données de référence, et notamment sur la mise en place d'une solution de MDM. **Néanmoins, cette démarche est complexe car transverse aux applications et nécessitant l'adhésion des métiers. Elle exige un appui managérial fort, ce qui est parfois difficile**.

On note deux types d'approches pour gérer les données de référence :

- **Approche tactique** (*bottom-up*). par l'apprentissage au niveau DSI. On reste dans un périmètre restreint et on commence par du prototypage de solutions du marché ou de développements spécifiques.
- **Approche classique** (*top-down*) par la conjugaison :

– de la mise en place d'une gouvernance d'entreprise (voir troisième partie du livre) ;
– du lancement de projets pilotes.

L'approche classique semble la meilleure, mais nécessite une forte implication des métiers. Sans cette implication, les DSI devront se rabattre sur une approche tactique puis embarquer les métiers peu à peu.

Il est essentiel de « **commencer petit mais voir grand** »**,** ce qui se décline à la fois dans le périmètre d'un projet et dans celui du SI.

Dans le périmètre d'un projet, il faut :

- définir un **périmètre restreint** mais offrant un gain sensible afin de faciliter l'adhésion ;
- **lotir afin d'assurer la courbe d'apprentissage** de l'entreprise.

Dans le périmètre du SI, il faut :

- penser et coordonner chaque projet **en accord avec une vision d'ensemble** (lié à des études d'urbanisme en général) ;
- mettre en place progressivement une **infrastructure mutualisée** et industrialisée.

Une autre caractéristique forte concerne la difficulté de la mobilisation des métiers :

- incompréhension quant à la valeur dégagée : les projets qui mettent en place une solution MDM offrent rarement un gain métier directement mesurable (sauf pour les projets de centralisation avec *re-engineering* du processus référentiel).
  Les apports doivent être analysés en pré-projet et mesurés en post-projet au travers des bénéfices indirects aux applications que le référentiel soutient (amélioration des indicateurs, amélioration des processus métier…).
- implication nécessaire : un accompagnement indispensable pour la compréhension des enjeux et la définition des objectifs.

8.5 BONNES PRATIQUES

Les tableaux qui suivent soulignent quelques bonnes pratiques pour la gestion des données, classées par :

- métier et urbanisme ;
- architecture.

Ces bonnes pratiques sont à considérer en général dans un projet qui inclut de manière directe ou indirecte la gestion des données de référence. Certaines peuvent être pérennisées, participant ainsi aux méthodes de gouvernance. Il ne s'agit pas ici d'être exhaustif, mais de décrire ce qui est essentiel.

8.5.1 Métier et urbanisme

Tableau 8.6 – Bonnes pratiques métier et urbanisme

Bonnes pratiques	Priorité */**/***
Identifier les données de référence.	***
Définir et, si possible, **décrire** les données de référence (glossaire sémantique). Exemple de définition métier d'un client : personne physique ou morale qui a été ou est susceptible d'être bénéficiaire d'un produit ou service fourni par l'entreprise.	***
Modéliser les données de référence (fédérer si nécessaire les différentes vues métier) : – Identifier les modèles et les formats existants qui s'appliquent. – Déterminer les identifiants. – Créer puis valider le modèle si aucun modèle existant ne s'applique (préférer les modèles simples avec liens sur d'autres modèles).	***
Spécifier la **qualité des données** à atteindre dans les processus pour la performance du métier.	**
Décrire les **processus référentiels** spécifiques aux données de référence.	**
Décrire le **cycle de vie métier** des principales données de référence. Exemple pour un client : prospect, client, ancien client… Identifier les règles de gestion associées. Exemple : événement qui fait passer un prospect à l'état de client.	**
Identifier si possible des sous-systèmes indépendants de gestion des données de référence (permettant par exemple de **centraliser l'acquisition de données de référence externes et de données paramètre).**	**
Spécifier les contrats d'échange des données de référence critiques (voir cette notion de contrat d'échange dans la troisième partie de l'ouvrage).	**

8.5.2 Architecture

Tableau 8.7 – Bonnes pratiques architecture

Bonnes pratiques	Priorité */**/***
Identifier les applications points de vérité pour les données de référence. Corollaire essentiel : toute application point de vérité d'une donnée doit permettre sa diffusion ou son interrogation.	***
Spécifier les **méthodes de diffusion des données de référence** (SOA, événementiel…).	***
Décrire le **cycle de vie technique** des données de référence (ex : qui crée le client…).	**
Déterminer la **qualité et la fiabilité des données des systèmes sources.**	**
Spécifier la **gestion de la qualité des données de référence** (où et comment gérer pour atteindre le niveau de qualité spécifié).	**
Déterminer les **profils (droits)** associés aux données de référence.	**
Spécifier la **gestion des métadonnées** : que décrire, où gérer, où stocker, comment publier et mettre à jour…	**

En résumé

Les critères de sélection d'une solution par rapport à une autre (MDM, développement spécifique, progiciel) dépendent de l'implication de l'entreprise dans la mise en œuvre de la solution et donc du caractère critique de celle-ci et du périmètre à couvrir.

Les exemples de mise en œuvre démontrent le besoin d'une démarche par étapes, induisant une phase d'étude importante afin de se projeter vers une cible au travers d'itérations. Le MDM doit être considéré comme une brique transverse mutualisée au sein d'un SI.

TROISIÈME PARTIE

Piloter : méthodes et organisation

9

Gouvernance des données de référence

Objectif

Nous allons définir la notion de gouvernance des données, sous-ensemble de la gouvernance d'un SI. À partir d'un modèle et d'un cadre de gouvernance intégrant les objectifs, les contraintes et les leviers d'action, nous proposons des exemples d'organisation (instances, rôles et acteurs), de règles et de procédures de gestion et d'outils de gouvernance.

9.1 DÉFINITION DE LA GOUVERNANCE DES DONNÉES

Selon le Gartner, la **gouvernance des données** est **une collection de bonnes pratiques** qui considèrent l'information comme une ressource à part entière de l'entreprise. Cette ressource doit être gérée avec des **règles précises, des processus et responsabilités clairement établis et véhiculés au sein des organisations dans le respect de leurs standards technologiques.** Il faut considérer la gouvernance informationnelle comme faisant partie intégrante de la gouvernance globale des organisations.

De manière plus pratique, **la gouvernance des données englobe tout ce qui permet de gérer de manière optimale les dimensions qualité, disponibilité, sécurité et conformité réglementaire de la donnée**.

La gouvernance des données reprend les principes déjà connus mais pas toujours bien pratiqués de leur administration. Elle inclut notamment des notions de pilotage qui permettent sa prise en charge par les métiers et étendent son spectre de pilotage au-delà des aspects purement techniques.

À la différence des données qui restent dans le périmètre d'usage d'un nombre limité d'applications, les données de référence transverses imposent une démarche toujours plus rigoureuse de gouvernance. Cette gouvernance doit prendre en compte des enjeux IT et surtout métier. Elle doit offrir son propre cadre d'analyse mais aussi de mesure de la performance suivant des dimensions partagées par tous (IT, métiers ou encore différents domaines métier d'un paradigme). Ainsi, analyse, définition, projection et mesure permettent d'initier un cercle vertueux, d'instancier une démarche itérative de pilotage.

La gouvernance a pour principal objectif la génération de valeur pour les métiers, d'abord par l'amélioration de la qualité des données et ensuite par l'enrichissement du spectre informationnel porté par les données ou corrélées grâce à elles (EIM, *Enterprise Information Management*).

Cet objectif revêt des aspects IT et génère certains problèmes qui doivent être évités :

- Perte de la traçabilité sur le contenu des modèles : on constate qu'une donnée est présente dans un modèle mais on ne connaît plus son sens ou sa légitimité…
- Divergence de modélisation d'un même objet : plusieurs modèles différents existent pour représenter le même objet, ce qui conduit à une multiplication des logiciels d'implémentation (un programme par type de modèle) et complique la maintenance, la documentation et le support.
- Divergence de valeurs : un même objet présente des valeurs différentes selon les applications.
- Mauvaise qualité des données (doublons, complétude, etc.).

Faire de la gouvernance des données un instrument de génération de valeur pour les métiers implique aussi certaines conséquences les concernant :

- alignement sémantique entre domaine, dotation d'un glossaire métier, définition des concepts ;
- définition précise des procédures de gestion ;
- définition des instances de contrôle et de gestion opérationnelle ;
- définition des rôles et responsabilités ;
- définition des objectifs et mesure de la performance ;
- définition des plans d'action et de correction…

La gouvernance de la donnée a donc pour objet la valorisation de la donnée dans le temps. Elle doit être envisagée en dehors des contraintes spécifiques d'un do-

maine d'activités. Elle répond en cela au caractère transverse et indépendant des processus de la donnée.

C'est un cadre de contrôle qualité visant à évaluer, gérer, exploiter, optimiser, contrôler, entretenir, protéger et finalement à valoriser le patrimoine de données des entreprises. Il est généralement admis que les sociétés disposant d'un plan efficace dans ce domaine sont capables de produire durablement des données plus précises, complètes et cohérentes sur leurs activités dans l'ensemble de leurs services.

Les avantages liés au déploiement d'un programme de gouvernance des données de l'entreprise sont nombreux, à la mesure des défis impliqués par sa mise en place.

9.2 MODÈLE DE DÉPLOIEMENT DE LA GOUVERNANCE

La figure 9.1 décrit ce modèle : les flèches y figurent la forte interaction entre les étapes (itératives et non ordonnées).

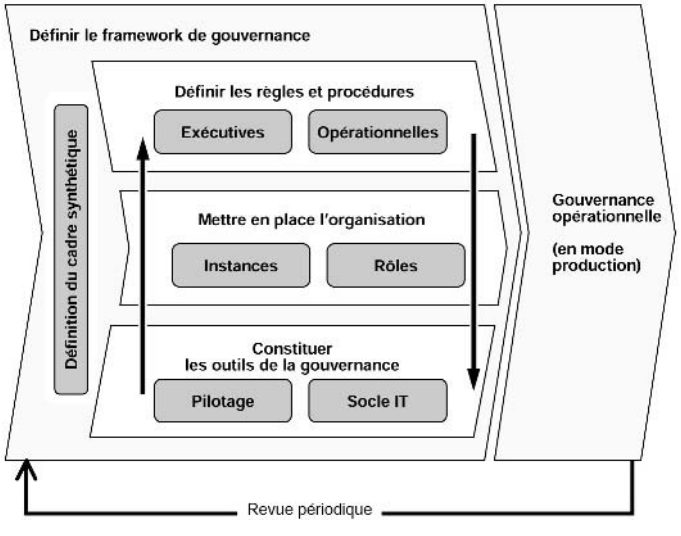

Figure 9.1 — Modèle de déploiement de la gouvernance

Définir le cadre de la gouvernance

Le cadre (*framework*) de gouvernance délimite les activités et concepts nécessaires à la mise en œuvre des projets référentiels. Son périmètre dépasse le cadre des projets et s'étend à l'organisation, assurant ainsi la pérennité des structures et des objectifs.

Il doit être considéré en premier lieu comme un cadre d'analyse mais aussi comme une cible. C'est un support des enjeux métier en cohérence avec la stratégie du système d'information. Dans sa forme simplifiée, il s'agit d'un cadre synthétique (que nous vous présentons plus loin).

La définition du cadre de la gouvernance englobe notamment les trois étapes suivantes.

Mettre en place l'organisation

La structure organisationnelle doit permettre de prendre en charge la gouvernance et le soutien des projets. Cette organisation s'étoffe au fil des projets et du déploiement de la gouvernance.

Définir les règles et procédures

Il faut édicter l'ensemble des règles et bonnes pratiques (ainsi que les éléments documentaires) nécessaires à la gouvernance. Ces procédures concernent d'abord les procédures « humaines » de gestion et s'étendent aux processus métier implémentés dans les outils. Elles s'appliquent aussi bien au niveau exécutif qu'au niveau opérationnel.

Constituer les outils de la gouvernance

Il faut constituer l'ensemble des outils et technologies mutualisables dans une démarche référentielle. Leur mise en œuvre se fait par étapes, au rythme des projets. On peut citer les outils suivants :

- description des données (métadonnées et dictionnaires) ;
- indicateurs de qualité des données et tableau de bord ;
- auditabilité des données (en particulier par rapport à des contraintes de conformité réglementaire et de sécurité).

Remarque sur l'étendue de la gouvernance

Si le cadre présenté ici a pour premier objectif l'assise et la gouvernance des données de référence, l'entreprise a intérêt à étendre ces recommandations au-delà de la simple sphère des solutions référentielles. Cette extension peut se faire selon deux axes :

- La gouvernance de la donnée de référence peut s'appliquer à l'ensemble du SI et pas seulement à son application support (le référentiel MDM). Cela signifie que le référentiel, les formats canoniques, les flux et l'utilisation des solutions doivent profiter du cadre de gouvernance.
- Ce modèle d'organisation peut s'étendre à d'autres données, surtout si ces données sont partagées. La gouvernance des données peut s'appliquer à l'ensemble de la sphère des applications transactionnelles, à celle des applications *middleware*, aux référentiels mais aussi aux ODS (*Operational Data Store*) et enfin à la sphère décisionnelle.

Il faut faire attention cependant aux embryons d'organisation existant souvent au niveau du décisionnel. S'il faut effectivement les inscrire dans le cadre de gouvernance, il faut se garder d'en faire le socle de la gouvernance. Pour une bonne gouvernance des données, la qualité et la cohérence doivent être introduites plus tôt dans la chaîne de l'information, ce qui n'est pas le mode natif de fonctionnement des équipes décisionnelles.

9.3 EXEMPLE DE CADRE SYNTHÉTIQUE DE GOUVERNANCE

Dans l'exemple suivant, le modèle de gouvernance (figure 9.2) comprend huit thèmes répartis en deux séries. La première série définit les objectifs et les contraintes (périmètre du cadre), la seconde les leviers d'action (centre du cadre).

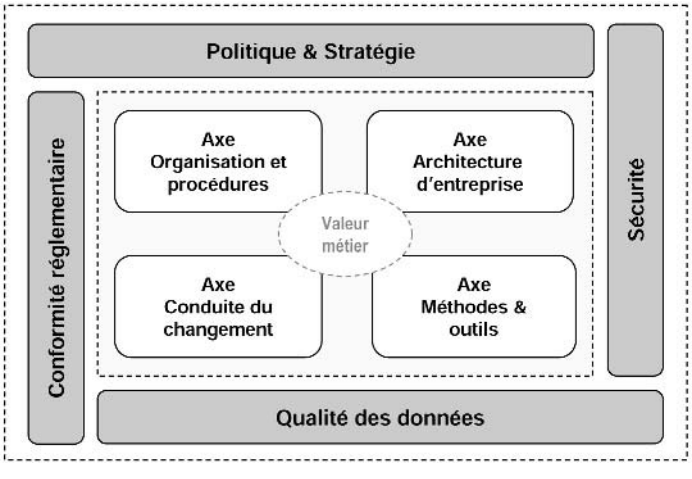

Figure 9.2 – Cadre synthétique de gouvernance

Objectifs et contraintes

- **Politique et stratégie** : association des référentiels à la stratégie d'entreprise et définition de la volonté stratégique d'agir et des moyens associés.
- **Sécurité :** définition des niveaux de sécurité et des procédures spécifiques associés à la donnée (auditabilité), sous forme d'instanciation de la politique globale de sécurité de l'entreprise.
- **Qualité des données** : définition des critères de mesure de la qualité de la donnée, des conditions d'obtention et de contrôle de cette qualité.
- **Conformité réglementaire** : injection dans ce cadre de gouvernance des contraintes externes qui réglementent l'accès, la sauvegarde, le pilotage des

données de l'entreprise, mais aussi participation de la donnée au respect de contraintes plus larges (notamment en termes de *risk & compliance*).

Ce sont ces éléments structurants qui définissent le cadre d'action de la gouvernance. En politique, on pourrait les comparer aux textes de loi qui définissent le cadre de liberté de toute action et, pour une vision dynamique, aux propositions de lois.

Leviers (moyens d'action disponibles)

Les leviers d'actions s'appliquent aux pratiques correctes de gouvernance des données. Ce sont les choix méthodologiques, organisationnels et techniques effectués par l'entreprise pour mettre en œuvre une bonne gouvernance. Nous les avons détaillés dans la deuxième partie du livre et nous les compléterons dans cette troisième partie. Ces leviers d'action sont révisables à une fréquence plus élevée que les objectifs et contraintes du cadre. Ils doivent être à tout moment en adéquation avec le cadre et la mise en œuvre de la gouvernance.

On distingue les leviers suivants :

- **Organisation et procédures** : il s'agit de définir à la fois les processus de gouvernance de la donnée et l'intervention des acteurs dans ces processus. Mettre en place une gouvernance a en effet un impact sur le rôle des acteurs impliqués.
- **Architecture d'entreprise** : positionnement des données dans un cadre plus large qui va de la prise en compte et de la modélisation des besoins et processus métier jusqu'à l'architecture. La contribution de chaque donnée, du point de vue métier, doit être identifiée et cartographiée puis déclinée sur son support de composants technologiques (notamment dans le cadre d'une stratégie SOA).
- **Méthodes et outils** : définition de l'outillage, du formalisme, des solutions techniques et méthodologiques associées à l'implémentation de la gouvernance. Définition de la cohérence technique multiréférentielle.
- **Conduite du changement** (compétences, communication, formation…) : elle est indispensable parce qu'une transformation, quelle qu'elle soit, rencontre toujours des obstacles !

9.3.1 Structure du cadre (objectifs et contraintes)

Politique et stratégie

La stratégie définit les **objectifs** assignés à l'entreprise et, au sein de ce cadre, leur impact au niveau des données.

Elle concerne une **vision à long terme.** Elle est suscitée par l'ensemble des opportunités et limitée par l'ensemble des contraintes. Son ambition est de perfectionner les valeurs intrinsèques et d'usage de la donnée (on parle d'« alignement » en urbanisme). **Appliquée à la donnée, la stratégie vise la valorisation de la donnée.**

La politique est la déclinaison factuelle et tempérée de la stratégie (déclinaison des axes de progrès en objectifs). Elle est réaliste et évolue dans le temps en fonction de la maturité de l'entreprise vis-à-vis de la vision portée par la stratégie. Elle implique une démarche d'acceptation et d'adoption par phase, car la formulation de chaque règle, étape ou processus doit répondre à la capacité d'adoption par les personnels.

Elle définit la **balance des pouvoirs** entre les acteurs impliqués dans la gestion des données et la gouvernance et préconise les degrés de responsabilité et d'organisation auxquels propositions, décisions, gestions et contrôles sont opérés.

Exemples de livrables associés :

- charte de gouvernance des données ;
- support documenté de la stratégie et de la politique associée ;
- *roadmap* stratégique ;
- fiche d'engagements signée par les parties.

Sécurité

Il s'agit de **définir les règles générales de sécurité pour tout type de donnée, et donc en particulier pour les données de référence**. Il faut fournir le cadre formel dans lequel doit s'inscrire le traitement des données afin de garantir le niveau de sécurité protégeant d'impacts légaux, financiers ou d'image. La sécurité sera notamment analysée à de multiples niveaux : attribut, objet, référentiel, multiréférentiel, ou encore en lien avec les processus amont ou aval. On peut par exemple laisser en libre accès la consultation de chaque fiche client d'un référentiel éponyme mais interdire l'extraction d'une liste de plus de cent fiches.

La sécurité peut influer sur la disponibilité des outils (par exemple, 24 h/24 – 7 j/7), isoler certains domaines métier (recherche et développement, finance), distinguer les différents niveaux de confidentialité, identifier des besoins de traçabilité, participer à la définition de la sécurité d'entreprise, participer à la définition des droits d'accès, à la gestion des risques, préconiser des rôles spécifiques (auditeur) ou encore imposer des processus et outillages (contrôle, audit).

Exemples de livrables associés :

- amendement « données » des documents sécurité d'entreprise ;
- analyse des risques et définition des niveaux de confidentialité des référentiels ;
- recommandation d'architecture ;
- définition des droits et responsabilités.

Conformité réglementaire

Les contraintes réglementaires sont issues d'un cadre légal national ou transnational. Des organismes sont en charge du contrôle de la conformité de l'entreprise au cadre réglementaire (CNIL, CRE, AMF, SGDN…). Par extension, des contraintes non

légales, librement acceptées par l'entreprise, peuvent aussi être incluses dans cette catégorie (par exemple une charte de respect environnemental ou une charte de respect des droits de l'homme pour les notations des agences de développement durable).

D'un point de vue fonctionnel et SI, ces contraintes réglementaires impactent généralement les données et notamment les données de référence, ainsi que les processus référentiels (création, modification, suppression, diffusion).

Par exemple : contraintes CNIL pour les clients « personnes physiques » et leurs droits d'accès et de correction des données détenues par l'entreprise, contraintes CRE (Commission de régulation de l'énergie) pour les « informations commercialement sensibles ».

Ces données étant réparties dans diverses applications, **mettre en place un référentiel permet d'en centraliser la gestion ainsi que l'auditabilité** du point de vue des instances de contrôle : instances internes qui utilisent les métiers et outils du contrôle interne et de l'audit et instances externes (autorités diverses).

Ainsi la conformité réglementaire impose la prise en compte de contraintes aux niveaux de la propriété des données (principalement des instances de données), de l'accès à ces données, de l'archivage et, plus généralement, toutes les opérations effectuées sur une donnée (exemple : CNIL sur traitement des fichiers de personnes physiques). Elle peut enfin imposer des obligations de *reporting* à une autorité externe.

Il faut donc lister les règlements connus et futurs (en prévision de leur entrée en application), déterminer les impacts sur les activités de la gestion des données puis, émettre les recommandations afin de satisfaire à la conformité réglementaire.

Exemples de livrables associés :

- répertoire documenté des lois et réglementations ;
- expressions de besoin, demande d'évolutions ;
- rapports auprès des autorités externes.

Remarque : on notera qu'en France, les entreprises de plus de cinquante employés doivent maintenant disposer d'un CIL (correspondant informatique et libertés) afin d'analyser chaque processus concernant des données « personnes physiques » et de satisfaire aux obligations de déclaration de la CNIL.

Qualité des données

La qualité des données couvre, comme nous l'avons déjà vu, des aspects intrinsèques et de services.

Gestion de l'alignement sémantique, maîtrise des modèles, gestion des ID et de l'unicité, dédoublonnage, gestion d'erreurs et autres sont autant de processus ou fonctions qui concourent à la qualité des données.

La qualité intrinsèque peut être outillée avec des outils de DQM.

L'amélioration dans le temps de la qualité d'usage repose, elle, sur l'analyse en masse et l'historisation d'indicateurs. Ces indicateurs sont soit communs à l'ensemble des référentiels, soit spécifiques à une donnée ou à un des métiers supportés par la donnée concernée : on trouve par exemple le taux de complétude moyen, les pourcentages de retours courriers pour cause de NPAI[1]…

Cette vision d'amélioration de l'usage peut aussi être outillée (outils de DQM, indicateurs et rapports).

La mise en œuvre de la gestion de la qualité passe par la définition d'un plan de qualité de la donnée. Ce plan est réparti en différents niveaux d'intervention et/ou séquencé selon les différentes étapes de la vie du référentiel :

- Analyse (*data profiling*) **:** analyse des données pour la recherche d'erreurs, incohérences, redondances de données et informations incomplètes. Comme évoqué, l'analyse est outillée par des outils de *profiling* qui permettent une analyse statistique soulignant les cas particuliers au travers des écarts.
- Nettoyage (*data cleansing*) : correction, standardisation et vérification des données.
- Intégration (*data integration*) **:** syntaxe canonique, réconciliation, et mise en relation de données sémantiquement liées.
- Enrichissement (*data augmentation*) : amélioration de la donnée par des sources internes et externes, dédoublonnage, fusion.
- Suivi (*data monitoring*) **:** suivi et contrôle de l'intégrité des données dans le temps.

Chaque étape peut donc être outillée et produire des rapports (voir figure 9.3).

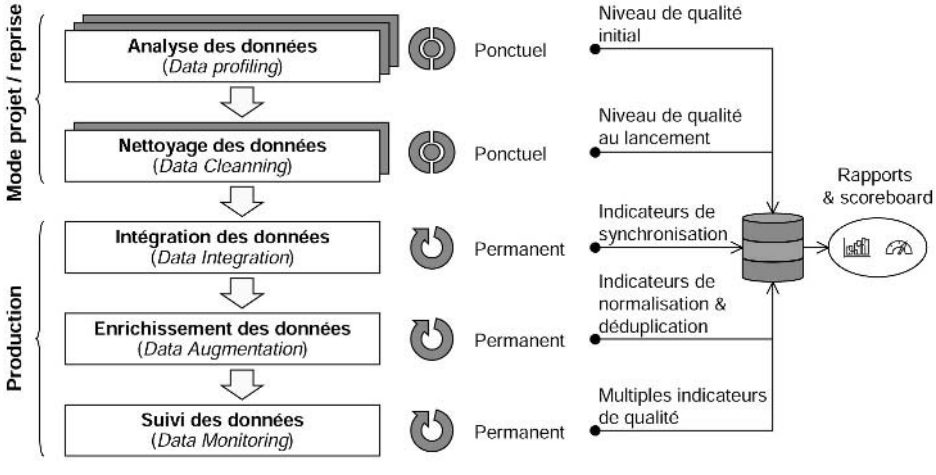

Figure 9.3 – Étapes de mise en œuvre de la gestion de la qualité

1. N'habite pas à l'adresse indiquée

Exemples de livrables associés :

- charte de qualité des données (principes) ;
- rapports d'audit qualité ;
- tableau de bord et suivi de la qualité ;
- *roadmap* qualité, matrice de pilotage et fiches actions.

9.3.2 Les leviers du cadre

Axe organisation et procédures

L'organisation à mettre en place pour répondre aux enjeux de la gouvernance des données doit dépendre du métier. C'est du moins l'objectif, car on constate fréquemment que les embryons organisationnels naissent plutôt dans la matrice IT. Il faut lutter contre cette tendance en favorisant le rôle de conseil et la mise en cohérence au sein de la sphère IT, tout en laissant la responsabilité de plein droit aux métiers.

L'organisation doit être strictement dépendante de la politique générale de l'entreprise en matière d'information. On peut distinguer cinq formes de politique[1] :

- l'utopie technocratique ;
- l'anarchie ;
- le féodalisme ;
- la monarchie ;
- le fédéralisme.

L'utopie technocratique se caractérise par une approche très technique, menant l'IT à définir, modéliser et catégoriser les informations en rapport direct avec l'outillage technologique. C'est le premier risque que nous dénonçons en cas de non-implication des métiers.

L'anarchie se définit par l'absence de politique d'entreprise en matière de gestion de l'information ou des données. Chaque individu agit en fonction de ses besoins (tableaux Excel…). L'intérêt du cadre de gouvernance et du MDM est justement de lutter contre cette situation et de favoriser la capitalisation et l'échange d'information.

Le féodalisme est un début de structuration de l'information au niveau d'un domaine métier ou d'une entité de l'entreprise pour l'usage quasi exclusif de ses managers. Cette forme archaïque de structuration, sans volonté de partage ou d'échange de l'information, émane généralement des premiers projets BI. C'est celle que nous découvrons le plus souvent lors de nos interventions en entreprise. Elle conduit chaque domaine métier à minimiser les problèmes, à en rejeter la responsabilité sur les métiers voisins et à ne pas considérer l'intérêt de l'entreprise en général.

1. Voir sur le sujet, l'excellent document « Information Politics » de Thomas H. Davenport, Robert G. Eccles et Laurence Prusak. *Sloan Management Review*, automne 1992.

La monarchie se définit par une approche globale, modélisée, dans laquelle les informations et obligations sont catégorisées pour chaque métier dans une optique de centralisation de l'information. On remarquera que ce type de politique est ordinairement mis en œuvre au bénéfice de la structure centrale et de ses dirigeants.

Le fédéralisme est le fruit de la négociation et du consensus. Issues de la volonté d'enrichissement croisé entre les domaines, la modélisation et la catégorisation de l'information sont définies en commun et échangées entre les participants.

Notre cadre de gouvernance vous poussera donc à abandonner l'utopie technocratique, l'anarchie ou le féodalisme pour, en fonction des particularités de votre entreprise, mettre en œuvre une monarchie ou un fédéralisme.

Cela peut se traduire par deux approches possibles :

- Une organisation plutôt centralisée : la gouvernance des données s'inscrit dans une logique de gouvernance métier et/ou SI. Les méthodes et l'organisation sont imposées, même si cela demande du temps et un accompagnement fort, voire des itérations sur les choix retenus.
  Si la DSI est puissante, le mouvement peut naître en son sein, mais attention à ne pas dériver vers l'utopie technocratique. Dans ce type d'entreprise, la fonction d'architecte des données est toujours reconnue et dotée de moyens importants mais il faut privilégier l'émergence et la prise de responsabilité du propriétaire de données.
- Une organisation plutôt décentralisée : la gouvernance est impulsée au croisement d'un besoin métier spécifique et d'une réflexion transverse (issue d'une cellule innovation ou autre prospective). Elle se met généralement en place à l'occasion d'un premier projet (démarche opportuniste).
  Il appartient alors au métier à l'origine de l'initiative d'embarquer les autres domaines en se souciant de l'intérêt de chacun et du bien commun. C'est le rôle du propriétaire de donnée, appuyé par son *sponsor*. Il revient à la DSI de supporter l'initiative, d'en assurer la cohérence en terme de normalisation et de technologie et de consolider les retours d'expérience afin de capitaliser et mutualiser les bonnes pratiques (ces dernières fonctions relevant de l'architecte de données).

Axe architecture d'entreprise

Cet axe regroupe la notion d'urbanisme et d'architecture (comme le terme anglo-saxon *Enterprise Architecture*). Le métier et sa prise en compte dans la structuration du SI sont importants. Les principaux objectifs sont :

- Identifier et cartographier les processus métier concourant aux données de référence (amont et aval).
- Définir et modéliser ces processus métier.
- Structurer l'alignement sémantique.

Il faut dériver la vision métier afin de la traduire au sein du SI :

- Identifier les données de référence et déterminer leur cycle de vie.

- Définir le positionnement des référentiels de données échangées au sein du système d'information.
- Établir des scénarios d'évolution fonctionnelle du SI.

On cherche à garantir la cohérence et la pertinence architecturale des projets dans le respect des bonnes pratiques de gestion des données de référence et des évolutions connues du SI (portefeuille projet).

Axe méthodes et outils

L'objectif est de donner aux projets les méthodes et outils permettant la mise en œuvre des solutions de gestion des données de référence.

Axe conduite du changement

Plusieurs cibles sont envisageables :

- Communiquer pour emporter l'adhésion des métiers à la démarche.
- Former les intervenants projets.
- Former les utilisateurs.
- Accompagner les projets.
- Supporter les intervenants en phase opérationnelle.
- Perpétuer les actions pour l'inclusion de nouveaux participants ou en mode correctif.

9.4 ORGANISATION

L'organisation se divise classiquement en trois niveaux :

- Le niveau stratégique : sponsors et encadrement directorial.
  – définit la stratégie et la politique des données ;
  – contrôle leur application ;
  – assure le financement ;
  – autorise et gère les modifications d'organisation liées à la donnée ;
  – valide les solutions et les projets.
- Le niveau exécutif et transverse : management.
  – développe et contrôle l'application des procédures, méthodes et bonnes pratiques ;
  – valide et centralise les définitions de données ;
  – assure la cohérence métier et anime les réseaux de propriétaires de données et de processus ;
  – est garant des normes et standards ;
  – soutient les projets.

- Le niveau opérationnel : utilisation, gestion, intendance et projet.
  – participe à la définition des données ;
  – définit les droits de gestion des données dans le respect des règles ;
  – assure l'alignement des projets avec les pratiques de gouvernance ;
  – assure la gestion de la qualité des données ;
  – assure la gouvernance opérationnelle de la donnée ;
  – déploie et gère les briques d'infrastructure (socle technique et de gouvernance).

On remarque que l'organisation est propre à chaque entreprise et oblige à s'adapter à son système politique général (monarchique, fédéral…). On définit donc pour chaque entreprise des instances regroupant les intervenants autour de la donnée. En fonction du contexte (mono ou multiréférentiel), du périmètre (domaines concernés), on peut définir une ou plusieurs instances (réparties par niveau d'organisation et/ou par domaine de responsabilités) ayant en charge la coordination des intervenants.

En première approche, on peut envisager la mise en place d'une **instance pérenne** (que l'on pourrait appeler **comité de gouvernance des données, comité de gestion des référentiels ou cellule centrale de coordination des référentiels)** qui arbitre entre les demandes effectuées par les projets et définit les règles, les organisations et outils transverses. Cette organisation, dans l'idéal, est mixte SI-métiers.

Les rôles spécifiques à la gouvernance des données ont un poids particulier dans ces instances.

9.4.1 Organisation de la gouvernance au niveau entreprise

La figure 9.4 schématise les rôles que nous allons définir et leur place dans l'organisation.

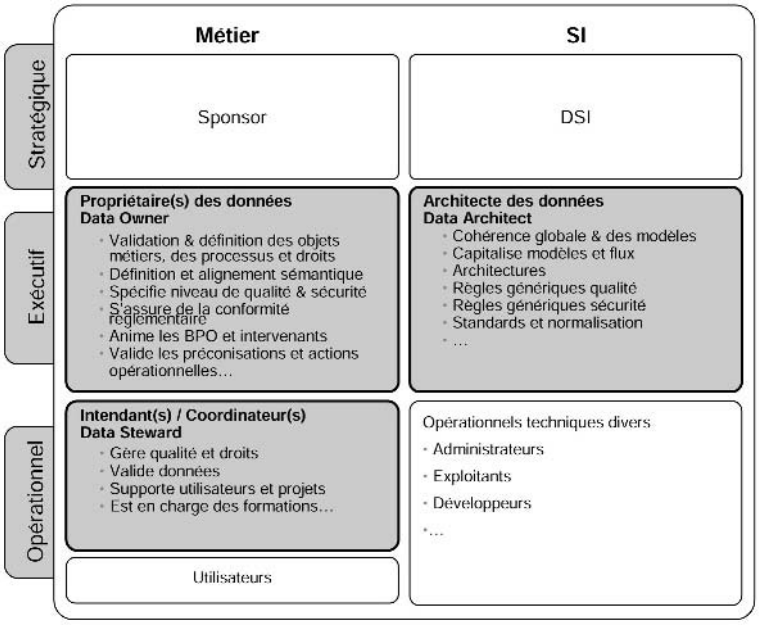

Figure 9.4 — Rôles définis pour la gouvernance des données

9.4.2 Rôles et acteurs

De nombreux acteurs interviennent au sein de l'organisation à divers titres, chaque acteur étant susceptible de remplir un ou plusieurs rôles.

> Attention à ne pas confondre rôles et acteurs : les rôles que nous définissons pourront être tenus par des individus ou des entités au sein des organisations existantes (il n'est pas forcément nécessaire de créer de nouveaux postes !).

Nous indiquerons s'il s'agit de rôles métier ou SI.

Nous ne détaillons pas ici les rôles classiques d'administration de données (administrateur technique, exploitant…).

On veillera lors de la mise en place du cadre de gouvernance à définir et/ou à compléter les fiches de poste des intervenants nommés pour remplir les rôles. Ainsi on pourra décliner des objectifs personnels annuels en rapport avec les objectifs du cadre de gouvernance.

Sponsor

On trouvera au niveau directorial la notion de **sponsor** : ce rôle est essentiel à la bonne mise en œuvre de la démarche de gestion des données de référence. Il définit et valide la stratégie de l'entreprise dans ce domaine.

> Le sponsor met à disposition les moyens (ressources et personnels), pilote les alignements stratégiques et s'assure du maintien de cette stratégie en mode opérationnel.

Propriétaire

Il s'agit nécessairement d'un intervenant ou d'une entité métier.

Le propriétaire **(*data owner*, *Business Data Owner*, ou BDO) définit les principales règles relatives à la création, modification et utilisation de la donnée (modèles, instances, services).** Il doit aussi piloter les éléments relatifs à la sécurité (droits d'accès, droits d'utilisation, sensibilité des données…) et à la qualité (limités à la définition du niveau de qualité requis pour les performances du métier).

Pour cela, **il interagit avec les propriétaires de processus métier (*Business Process Owner* ou BPO)** dont les données participent aux processus amont ou aval du référentiel. Le propriétaire prend en compte leurs besoins et aligne les usages et la prise en compte des données de référence suivant les règles de gouvernance. En ce sens, le rôle du propriétaire s'étend au-delà du périmètre de gestion du référentiel. Il assume une véritable fonction transverse sans pour autant se substituer aux responsabilités des propriétaires de processus métier. Un travail coopératif, en réseau, s'avère être le plus productif.

Le propriétaire est responsable de la définition de la structure de l'objet pour le compte d'un domaine métier (sauf pour les données acquises à l'extérieur). Il en décrit si nécessaire les métadonnées (définition sémantique, syntaxe des attributs

utilisables, règles de gestion associées…). Dans le cas où les progiciels imposent les modèles, il travaille par alignement (*mapping*).

Dans nos propositions de rôles, le propriétaire a en charge l'animation et le management d'ensemble. Cependant dans un contexte multiréférentiel ou pour un référentiel d'entreprise important, on peut redistribuer ses responsabilités.

Propriété des modèles

Les objets et paradigmes largement partagés peuvent nécessiter la participation des plusieurs propriétaires, qui auront autorité sur les attributs de leurs domaines.

Plusieurs propriétaires peuvent intervenir dans la définition des objets partagés au sein des référentiels, en lien avec l'architecte de données (voir ci-après). La gestion de la multipropriété implique la mise en commun d'un « noyau d'attributs » puis la **séparation de l'objet suivant chaque dimension métier soutenue (en ce sens il peut y avoir autant de propriétaires que de métiers concernés)**.

On pourra nommer un « directeur des données » (sorte de « super propriétaire ») ou on définira une entité transverse (sorte de « syndic de propriété ») pour arbitrer si nécessaire. En dernier ressort, il sera détenteur de la propriété pour le compte de l'entreprise et non plus de tel ou tel métier. Ainsi les attributs communs seront sa propriété.

Rappelons que si la multipropriété a un sens au niveau d'un objet métier, **un attribut a cependant obligatoirement un et un seul propriétaire**.

La figure 9.5 schématise, à titre d'exemple, les rôles de propriétaires concernant la modélisation de l'objet « compte » partagé entre les métiers de la gestion et de la comptabilité.

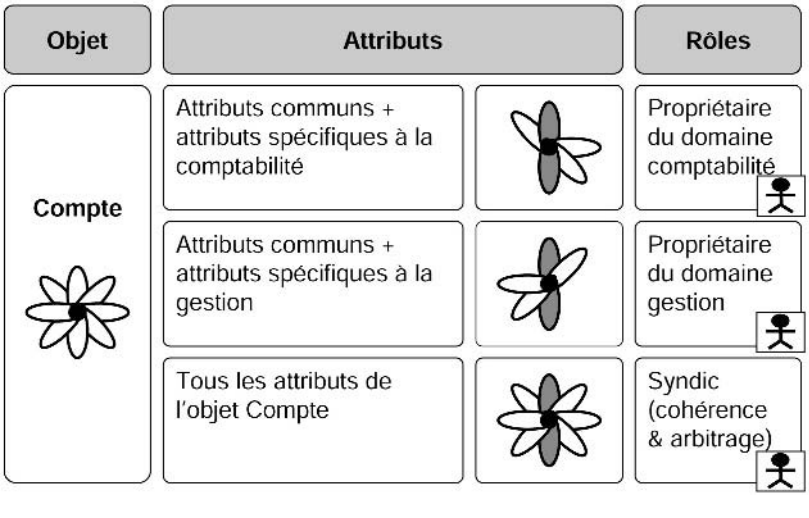

Figure 9.5 – Propriétaires des modèles pour l'objet compte

Propriété des instances

Le propriétaire des instances est celui qui définit les droits de création ou de modification des données pour les attributs d'une instance, en ce sens c'est en général la même entité que le propriétaire des modèles. **Le propriétaire délègue ses droits aux utilisateurs qui créent et modifient les instances à l'intérieur d'applications**.

Un intendant ou coordinateur (voir ci-après) peut éventuellement valider les données au niveau de l'application « point de vérité ».

> L'identification du point de vérité est essentielle pour déterminer le propriétaire des instances.

La figure 9.6 illustre le rôle des utilisateurs (avec les droits spécifiés par les propriétaires) dans la création et la modification des données, avec validation éventuelle par le coordinateur.

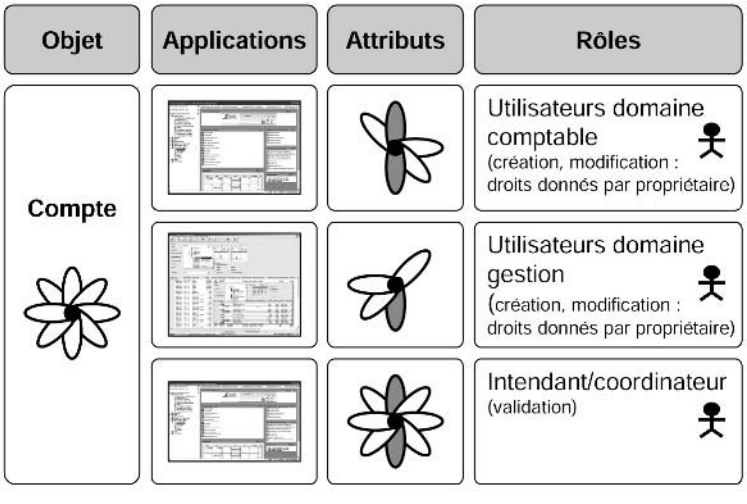

Figure 9.6 – Gestion des instances pour l'objet compte

Coordinateur (ou intendant)

Il s'agit de préférence d'un intervenant métier, mais, exceptionnellement un intervenant SI peut aussi tenir ce rôle par délégation.

Le coordinateur **(*data steward*,** que l'on peut aussi traduire **par intendant) est en charge, au quotidien, de la gestion des données. Il gère opérationnellement la qualité de la donnée et supporte la mise en œuvre des règles de gouvernance et des décisions du comité**.

En mode projet, il participe à la spécification des solutions sous la responsabilité du propriétaire (modèle de données et droits d'accès et de diffusion, critères qualité

requis) prépare les indicateurs de suivi puis gère la mise en œuvre au sein des solutions.

En production, il soutient les utilisateurs et il peut être également chargé de la validation des données créées, de leur diffusion ou encore être responsable de certaines interventions (dédoublonnage).

Il encadre en permanence l'utilisation du référentiel et doit être un référent connu pour l'ensemble de l'entreprise.

Utilisateur

Ce rôle est rempli par les intervenants métier sur la donnée ou délégué à des maîtrises d'œuvre.

L'utilisateur peut avoir des pouvoirs plus ou moins importants en termes de périmètre ou d'action (selon les règles définies par les propriétaires et mises en pratique par les intendants/coordinateurs).

La délégation des mêmes pouvoirs à un partenaire est possible.

On identifie deux types d'utilisateurs, les utilisateurs actifs (ayant des droits type création/modification/suppression) et les utilisateurs passifs (disposant de simples droits de consultation).

Comme pour les propriétaires du modèle, on note que de multiples intervenants peuvent être en charge de la complétude d'une occurrence d'objet. Même si ces cas ne posent pas de problèmes théoriques, la multi-utilisation sur un attribut d'une même occurrence d'objet doit répondre à des règles simples. La plus simple des règles étant « pas de recouvrement », on favorise un périmètre de responsabilité fixe pour chaque intervenant/rôle de manière à ne pas générer de conflit, notamment entre métiers (on peut se caler sur le périmètre de propriété des attributs des modèles).

Architecte de données

Il s'agit d'un intervenant SI.

Il conseille les propriétaires pour la modélisation. Il valide la cohérence des modèles, à l'échelle d'un paradigme mais aussi pour l'interopérabilité entre de multiples paradigmes (ce point est essentiel car il est aussi générateur de valeur pour l'entreprise grâce au croisement de données qui peut en résulter). Il s'assure que les objets ne comportent ni redondance ni incohérence d'un métier à l'autre et que les définitions sémantiques de chaque attribut soient documentées et respectées. Il s'attache aux respects des normes, standards et règles de sécurité. Il participe à la définition des règles et procédures de gestion.

Il participe à la définition architecturale des solutions (architecture applicative et technique). Il bénéficie d'une bonne maîtrise des méthodes agiles (UML…)

En termes de profil, il a une compréhension des processus métier qui manipulent les données et connaît les applications correspondantes. Il maîtrise les outils de modélisation, les technologies de gestion et de qualité des données.

Fiches résumés des rôles

Les fiches qui suivent résument les principaux rôles évoqués : propriétaire, intendant/coordinateur, architecte des données. Nous précisons en particulier les missions, responsabilités et compétences requises.

Propriétaire – ***Business Data Owner***
Niveau organisationnel
Exécutif
Missions
Le BDO coordonne l'ensemble des acteurs (BPO, intendant, *data architect*…) et veille à la mise en œuvre des directives du comité de gouvernance. Il est le garant des définitions sémantiques et de la cohérence structurelle des données avec le concept métier. Ses responsabilités sont en rapport avec la donnée et aussi transverses (périmètre du référentiel).
Responsabilités
À ce titre, il a en charge : – la définition des règles d'éligibilité des données au référentiel ; – la nomination et le management des intendants ; – le suivi des indicateurs ; – la validation des évolutions importantes ; – la coordination des acteurs ; – les arbitrages exécutifs ; – la gestion des évolutions fonctionnelles de l'outil mis en œuvre ; – la définition des concepts (tache sémantique) et du périmètre de la donnée ; – la définition, validation des spécifications des objets et services associés ; – la cohérence entre structure et concept métier ; – la conformité réglementaire de la donnée et de ses traitements ; – le traitement des demandes d'évolution structurelle ; – la communication globale sur la donnée ; – la prise en compte de la donnée par les BPO et leurs applicatifs.
Compétences requises
Fédération d'équipe. Animation collaborative. Maîtrise du périmètre des métiers de l'entreprise. Sensibilité à la stratégie SI. Maîtrise des processus de décision. Expérience du pilotage de projet. Expertise métier sur les données dont il a la responsabilité. Vision globale des utilisations qui en sont faites. Capacité à se coordonner avec ses homologues sur des sujets frontières.

Intendant/Coordinateur
Niveau organisationnel
Opérationnel
Missions
Le coordinateur fait le lien avec les utilisateurs. Dans un sens, il veille à la prise en compte de leurs besoins et remonte demandes et analyses au propriétaire. Dans l'autre sens, il veille à la prise en compte opérationnelle des règles et décisions émises par le comité de gouvernance. Ses missions d'intendance visent à la bonne utilisation du référentiel et de la donnée.
Responsabilités
À ce titre, il a en charge sur son périmètre métier : – la reformulation et la prise en compte des demandes ; – la relation utilisateur notamment la communication des décisions du comité ; – l'accompagnement des nouveaux utilisateurs et en particulier leur formation ; – l'accompagnement des projets mettant en œuvre la donnée ; – le support aux utilisateurs ; – la rédaction (ou participation à la rédaction) de la documentation ; – le suivi des fournisseurs tiers de données ; – les opérations de gestion avancées (par exemple déduplication) ; – la veille sur les pratiques de gestion de données ; – l'ouverture des droits (définition de groupe, rôle et droits) ; – la collecte des indicateurs qualité et la réalisation des tableaux de bord.
Compétences requises
Maîtrise de la cartographie des utilisateurs et des partenaires. Connaissance de la politique du/des propriétaires concernant les principales demandes standard. Compréhension des orientations et des contraintes en matière d'architecture et de qualité de la donnée. Capacité à nouer une relation pérenne de confiance avec les utilisateurs. Parfaite maîtrise de l'utilisation des outils et suivi de leur évolution.

Architecte des données
Niveau organisationnel
Exécutif
Missions
L'architecte des données a une vision globale des modèles de données et de l'architecture. Il conseille les responsables métier (dont les propriétaires). Il est le garant de la cohérence structurelle du référentiel.
Responsabilités
À ce titre, il a en charge : – la modélisation physique des objets ; – la mise en cohérence interne et inter modèle (coordination des métadonnées) ; – la mise en cohérence des modèles hors référentiel (flux, format canonique, décisionnel…) ; – la capitalisation des modèles ; – la capitalisation des définitions sémantiques ; – la coordination sémantique ; – la résolution et l'anticipation des conflits autour des données et la détection des opportunités d'amélioration (exemple : mise en cohérence, évolution des référentiels, amélioration de la qualité…) ; – la définition et la validation de l'architecture ; – les relations éditeurs ; – le support aux propriétaires de données et autres responsables (ex. BPO) et donc l'aide aux métiers pour appréhender la gestion des données et les technologies associées.
Compétences requises
Urbanisation des SI. Modélisation des données et méthodes agiles. Gestion des données maître (modèles et fonctionnalités). Gestion des contenus immatériels (sémantique, métadonnées). Architecture orientée services. Expérience du portefeuille applicatif. Capacité de coordination multiculturelle.

9.4.3 Mise en œuvre des rôles en pratique

Dans beaucoup de projets, le choix d'un progiciel métier impose les modèles des objets métier. Le rôle de **propriétaire** reste toutefois indispensable pour définir la sécurité et la qualité attendue des données, sélectionner les attributs réellement pertinents dans le contexte de l'entreprise et définir leur sémantique. On pérennisera au plus tôt ce rôle afin d'en faire le vecteur d'expansion des principes de gouvernance.

Le rôle de **l'intendant/coordinateur** peut être tenu par un utilisateur de l'application point de vérité.

Dans le cadre d'un projet, un urbaniste peut assumer le rôle **d'architecte des données**. Idéalement, cette fonction devrait être stable afin d'assurer une cohérence multiprojet.

Encore une fois ces rôles sont des rôles types, à reconsidérer au sein de chaque entreprise. Celui de propriétaire peut par exemple être réparti entre plusieurs personnes/rôles, en séparant, par exemple, les aspects « modélisation métier » de ceux de « manager de données ». Intendance et coordination peuvent aussi donner naissance à deux rôles en séparant les aspects de gestion centrale de ceux de support aux utilisateurs.

9.5 RÈGLES ET PROCÉDURES

Il s'agit de **définir les règles génériques et les procédures de gestion** (qui peuvent se décliner en consignes).

En première approche, ces règles et procédures peuvent être simplement liées au référentiel, donc relatives au cycle de vie métier et au cycle de vie technique de la donnée et, ainsi, pour chaque donnée de référence identifiée, il importe *a minima* :

- d'identifier l'application point de vérité (donnée valide) dans l'existant et la cible ;
- **de définir les conventions à utiliser (**vocabulaire, dénominations**…) si cela n'a pas été déjà fait ;**
- de vérifier que les procédures de gestion sont bien définies et documentées **et éventuellement déclinées en consignes**.

Une approche plus systématique et élargie du cadre de gouvernance reste néanmoins préférable, menant à une cartographie des activités liées à la gestion des données, du point de vue métier comme du point de vue IT (voir figure 9.7).

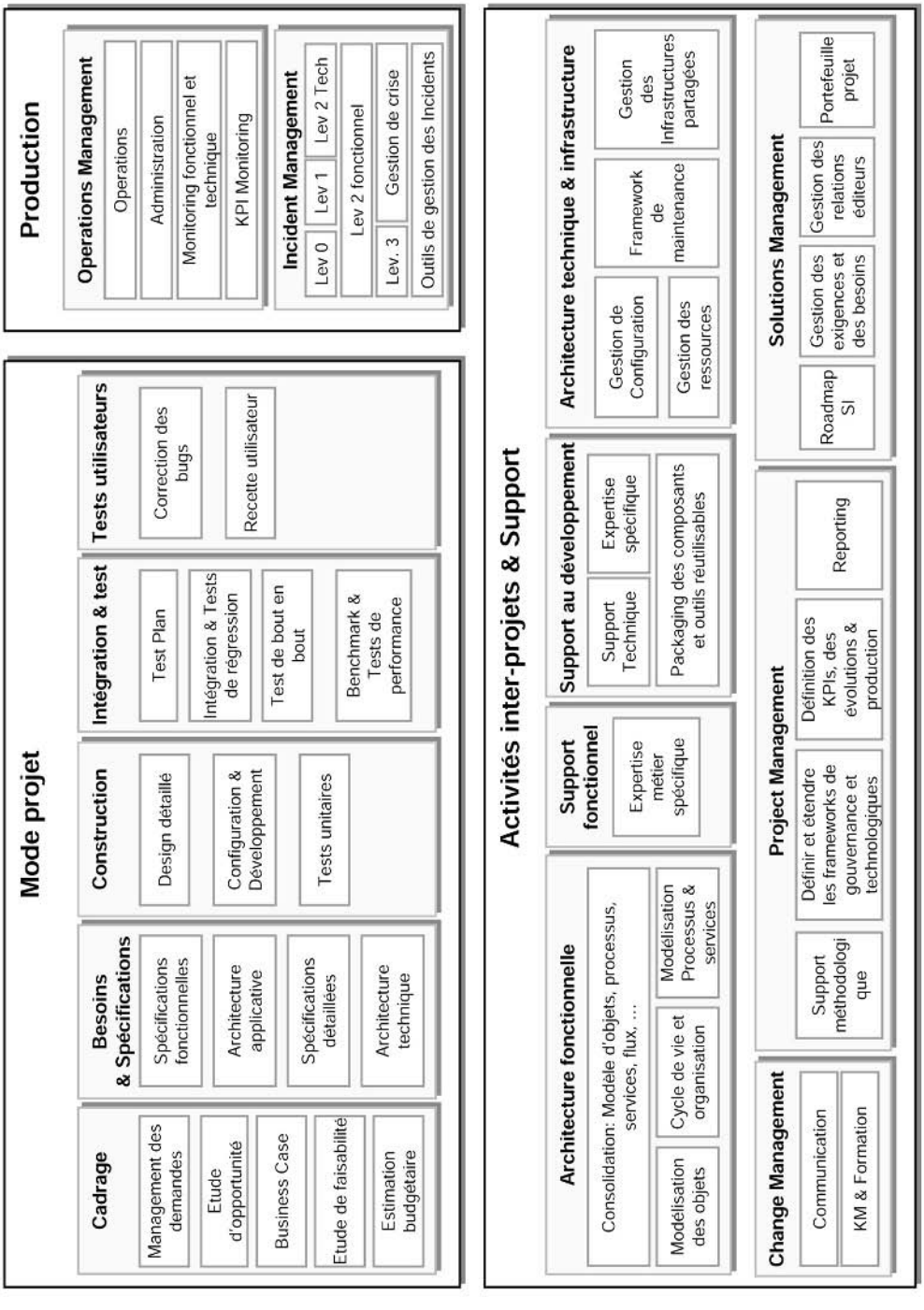

Figure 9.7 — Cartographie des activités

Cette cartographie peut ensuite être dérivée en processus et procédures outillées ou humaines, précisément définies, pour lesquelles on modélise le processus en incluant les intervenants (voir figure 9.8).

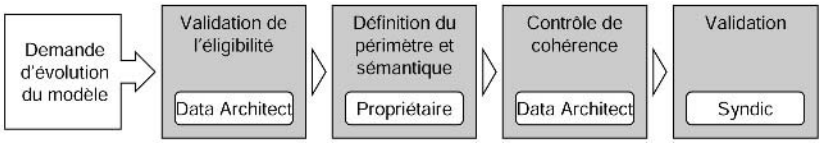

Figure 9.8 — Exemple de procédure simple de gouvernance

Enfin, l'ensemble de ces processus et procédures peut être croisé avec les rôles afin d'obtenir une matrice de responsabilité (RACI) (voir exemple ci-dessous, figure 9.9). Cette matrice est un bon indicateur de la balance des pouvoirs au sein de l'organisation.

	Rôles & équipes / Processus/procédures	Act	Métier				SI	
			Syndic	BDOs	Intendant	Utilisateur	Data Architect	Architecte technique
Urbanisation	Niveau de gestion & éligibilité	S/T	R	C			A R	C
	Analyse sécurité	P	R	C	I		C	A
	Définir les matrices de propriété	P	A R	C	C	I	R	C
	Définir l'architecture	S/T	I				R	A C
Modélisation des objets	Analyse des demandes et exigences	P	A	C			R	C
	Définition physique des objets	P	R	C	I	I	A R	C
	Définition fonctionnelle et sémantique	P	A R		I	I	R	C
Modélisation des processus	Analyse des demandes et exigences	P	A R				C	R
	Définir services d'acquisition	P					R	A
	Définir services de diffusion	P					R	A
	Indicateurs de processus	P	A R		C		C	R
	Définition fonctionnelle des processus	P	A R		C		C	R

A Accountable : Garant
R Responsable : en charge de
C Consulté
I Informé
S / T : Soutient / Transverse
P : phase Projet
R : production (RUN)

Figure 9.9 — Matrice de responsabilité

9.6 OUTILS DE GOUVERNANCE

9.6.1 Les outils de pilotage de la gouvernance

La gouvernance est un cadre, tout à la fois un cadre d'analyse et un cadre d'objectifs. En matière de gouvernance, « voir grand et commencer petit » est souvent la

conduite préférable à tenir. Nous rappelons que la maturité de l'entreprise face aux données de référence et à leur gestion est une réalité à prendre en compte avant de définir des objectifs inatteignables.

Pour gouverner, nous invitons chaque entreprise à décliner le cadre sous forme d'une « matrice de maturité » dont elle déduira les actions à mener (voir figure 9.10). Cette matrice peut être revue périodiquement afin de définir de nouveaux objectifs, de nouveaux paliers à atteindre. Elle peut se doubler d'une matrice spécifique à la qualité des données, que nous ne décrirons pas ici.

Leviers	Niveau 0	Niveau 1	Niveau 2	Niveau 3
Organisation	• Organisation non spécialisée • Pas de processus de gouvernance	• Organisation en charge de la gestion des données (collecte & validation) • Processus de création des données	• Responsabilités partagées entre les acteurs métier et IT • Modélisation des règles et processus d'éligibilité	• Comité de gouvernance inter-départements métiers • Optimisation du processus de modélisation des données
Architecture d'entreprise	• Données de référence dans chaque application métier • Pas de cartographie des données	• Une des applications métier est reconnue comme étant référentielle • Identification et localisation des données, des applications sources et cibles	• Existence d'un entrepôt référentiel • Définition des processus et des règles métier applicables à l'objet donnée	• Gestion centralisée des référentiels pour tous les systèmes • Définition des liens sémantiques entre les données
Méthode & outils	• Le modèle de données dépend des applications métiers • Pas de gestion de la qualité des données • Pas de sécurité • Toutes les informations sont accessibles	• Modèle de donnée central et consolidé des données de référence • Gestion basique de la qualité durant le processus de chargement des données • Journalisation (logging) basique	• Enrichissement du modèle avec les standards internationaux (hiérarchie, champs) • Amélioration intrinsèque et continue de la qualité dans les processus • Vues du référentiel restreintes par rôle	• Le référentiel supporte les formats canoniques standards de données de référence • Suivi des journaux-logs • Auditabilité (champs critiques)
Accompagnement au changement	• Pas de compétence dédiée à la gestion des données.	• Compétence locale sur les outils et les méthodes par les équipes techniques	• Mise en œuvre de processus de gestion des données de référence par métier.	• La gestion de la donnée est considérée comme un enjeu stratégique de l'entreprise par tous les métiers

Fiches Action

Figure 9.10 — Matrice de maturité d'entreprise

Ainsi, pour chaque levier, on peut identifier les actions à mener pour passer d'un niveau de maturité *n* à *n + 1*, les actions se déclinant sous forme de fiches d'actions.

9.6.2 Les outils technologiques relatifs à la gouvernance

Si la gouvernance est d'abord organisationnelle, elle repose aussi sur un outillage technologique que nous intégrons au socle des référentiels. Les outils technologiques envisageables sont (liste non exhaustive) :

- dictionnaire sémantique des concepts et données (*business glossary*) ;
- descriptif des métadonnées (répertoire des modèles et flux) ;

- indicateurs de suivi de la qualité des données ;
- traces et audit des données, etc.

9.6.3 Les priorités dans la mise en œuvre

La priorité est de définir le cadre et la matrice de pilotage (constituée à partir de la matrice de maturité et des fiches actions).

Puis il convient de soigner la sémantique et les modèles des données de référence identifiées. Pour cela, on peut envisager la mise en place d'un annuaire (*repository*) ou d'un outil de modélisation de type MEGA, PowerAMC (encore appelé *Power Designer*), Case Wise ou Telelogic qui tendront à couvrir la chaîne, des besoins métier à la consolidation des métadonnées.

Néanmoins, il peut aussi être très utile de se préoccuper du suivi de la qualité des données, en envisageant la mise en œuvre d'un outil de DQM permettant de générer des rapports sur les critères de qualité retenus.

9.7 EXEMPLES DE MISE EN PLACE DE LA GOUVERNANCE DANS DES ENTREPRISES

9.7.1 Un grand distributeur

La société X s'est dotée de principes de gouvernance de base, afin de répondre aux enjeux métier (qualité des données, qualité des processus pour améliorer les relations client fournisseur et diminuer les coûts de gestion). **Une instance centrale (le centre de compétence référentiel) est en charge de la maîtrise de la gouvernance (surveillance des pratiques et usages) ainsi que de la maîtrise technique et des évolutions**. Cette instance allie technique et métier, bien que le métier soit largement prioritaire dans le contexte de la grande distribution en général et de la société X en particulier.

Deux missions principales sont donc à la charge du centre de compétence :

- Gestion centrale des solutions :
  - support à l'alignement et aux déploiements locaux ;
  - définition du modèle de données et support aux intervenants GDS (standard international d'échange de donnée produit).
- Suivi de l'activité :
  - indicateurs de performance ;
  - indicateurs qualité ;
  - indicateurs de gestion.

Du point de vue métier, une grande liberté est laissée aux entités locales. Le centre de compétence étant plutôt porteur des bonnes pratiques mais ne pouvant répondre à toutes les spécificités à l'échelle mondiale.

9.7.2 Un producteur

La société Y a défini un nouveau plan d'urbanisme ainsi qu'un schéma directeur afin de refondre son système d'information. Elle s'est alors dotée d'une organisation en charge de la donnée. Cela passe essentiellement par la mise en place d'une organisation spécifique pour les données : la fondation des données de référence.

Un comité stratégique couvre l'ensemble des orientations SI de l'entreprise mais n'est pas spécifique à la donnée (comité de gouvernance). La société Y a engagé la mise en place de son cadre de gouvernance en commençant par la définition du cadre en tant que tel et des principales règles applicables aux projets. Elle s'est dotée et continue de se doter d'outils de consolidation au niveau de la fondation : dictionnaire sémantique des données, descriptif des métadonnées, référentiels de paramètres, *scoreboard* et indicateurs de suivi des référentiels.

Cette société, bien que multinationale, est centralisée quant à ses instances de décision. La mise en place d'un cadre de gouvernance pour les données de référence a conduit à une réflexion sur l'extension des principes à l'ensemble des données. La fondation des données de référence est naturellement devenue la fondation des données (*data foundation*).

En résumé

La gouvernance des données est une déclinaison de la gouvernance d'entreprise : gouverner, c'est **piloter** et **contrôler**.

Le pilotage s'appuie sur la mise en place d'une organisation adéquate et, en particulier, sur l'identification de rôles parmi lesquels celui essentiel de **propriétaire des données**. Celui-ci définit les principales règles relatives à la création, la modification et l'utilisation des données (modèles et instances). Elles sont placées sous sa responsabilité.

Au niveau SI, un **architecte de données** garantit la cohérence et la capitalisation des travaux réalisés sur les données.

Le contrôle consiste à spécifier et maintenir le niveau adéquat de **qualité des données**, *via* éventuellement un outil de DQM.

10

Étapes de déploiement d'un projet de gestion des données de référence

Objectif

Il s'agit ici de mettre l'accent sur les particularités de tels projets :

– tâches spécifiques par rapport aux tâches classiques de tout projet SI ;
– éléments de méthode ;
– caractéristiques remarquables des charges de mise en œuvre par rapport à la moyenne des projets.

De par la transversalité induite par de tels projets, la démarche de mise en œuvre d'un projet de gestion de données de référence (ou de la prise en compte de la gestion des données de référence dans un projet) comporte donc certains points spécifiques (urbanisme, architecture, spécifications) ou demande une attention particulière sur certaines étapes (tests, intégration, conduite du changement). C'est ce que nous allons principalement décrire. Nous finirons par indiquer les principales bonnes pratiques en terme de méthodes et d'organisation.

10.1 PRINCIPES GÉNÉRAUX

Notre propos n'est pas ici de vous exposer une démarche projet détaillée. Le spectre est en effet bien trop large entre les différents types de projets MDM (PIM, CDI, autres…) et les différents modes d'implémentation ou encore l'outillage choisi (solution MDM, développement spécifique, progiciel métier).

Nos indications portent sur une initiative multiréférentiel (par exemple pour l'étape « identification des données ») puis se déclinent en sous-projets par donnée.

Notons que ces sous-projets créent des livrables légèrement différents en fonction de l'outillage. En effet, si l'outil choisi comporte déjà un modèle de données ou possède des processus implémentés, on travaille par **alignement** (*mapping*). Dans le cas contraire (pas de modèle, pas de processus) on travaille par **modélisation**.

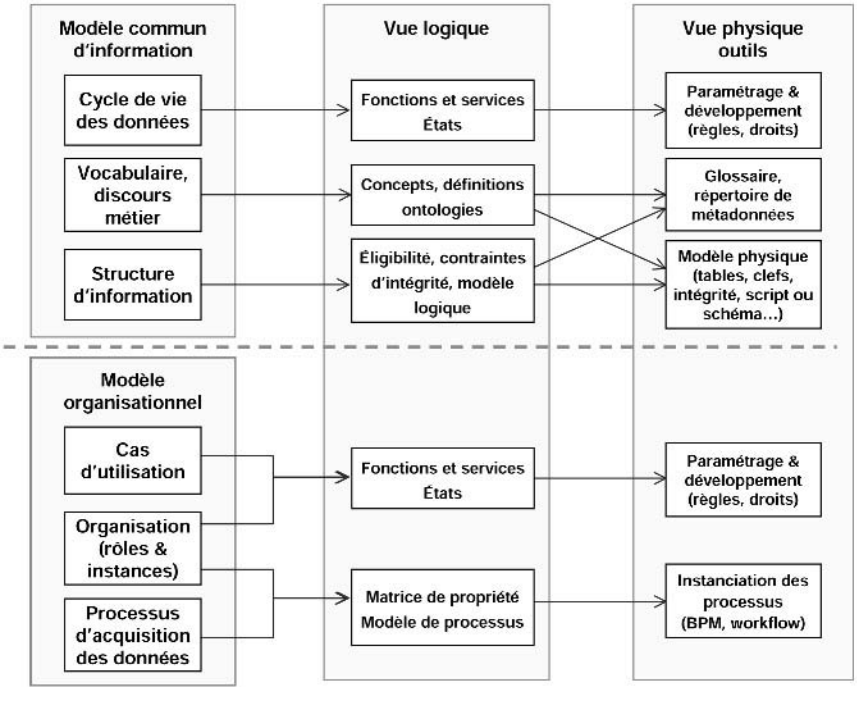

Figure 10.1 — Vue synthétique et générale de modélisation/spécification d'un projet référentiel[1]

Remarquons enfin que les spécifications de tels projets ne doivent pas être spécifiques à un domaine métier en particulier. Cela signifie que vous devez prendre en

1. Notre vue synthétique s'inspire librement des travaux de Pierre Bonnet au sein de l'initiative « MDM Alliance Group », initiative à laquelle participent certains des auteurs. (http://www.sustainableitarchitecture.com/mdm_alliance).

compte chaque besoin ou exigence sans pour autant générer une adhérence propre à tel ou tel domaine. Contrairement aux modèles de gestion dépassés des bases de données, nous nous intéressons à la transversalité de la solution, à son intégration dans les différents processus de l'entreprise. L'alignement préconisé concerne plus l'organisation et la sémantique.

Vos projets doivent avant tout prendre en compte **les processus** et **cas d'utilisation** de la donnée, vos **spécificités organisationnelles**, les **notions de pilotage associées à la gouvernance** et les contraintes **d'intégrité multi-référentielles** avant d'aboutir à un modèle de donnée et une architecture (voir figure 10.1).

10.2 TÂCHES SPÉCIFIQUES À LA GESTION DES DONNÉES

La figure 10.2 identifie les tâches spécifiques liées aux données de référence par rapport aux tâches « classiques » d'un processus projet. Ce sont les tâches principales que nous allons décrire.

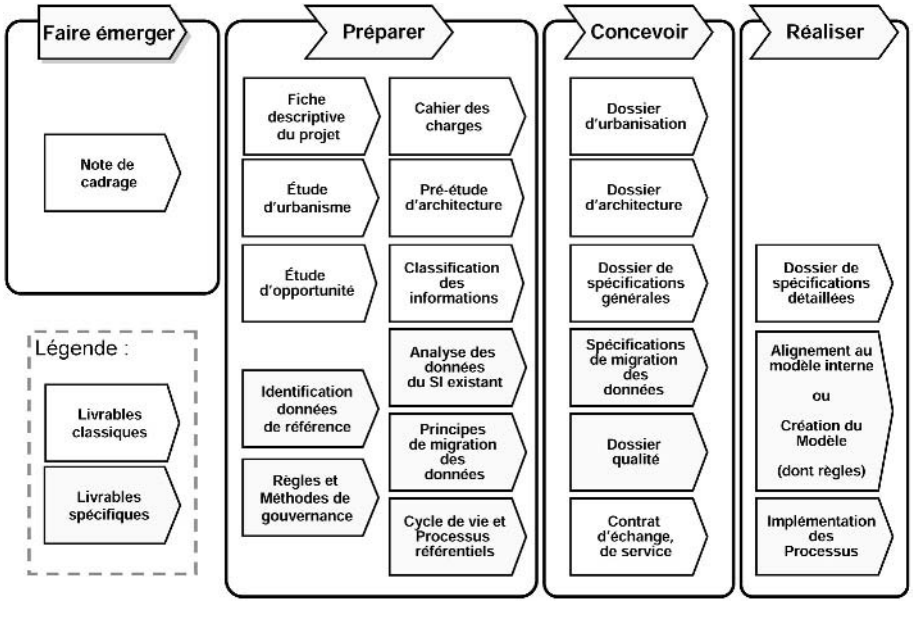

Figure 10.2 — Tâches spécifiques liées à la gestion des données de référence

10.2.1 Identifier et décrire les données de référence

Cela consiste à identifier les données de référence à prendre en compte, en partant des objets métier les plus structurants.

Rappelons qu'une donnée de référence est partagée par plusieurs processus (et donc plusieurs applications)**, ce qui est un critère d'identification important**. La description de ces données (à quoi servent-elles, comment sont-elles employées ?) nécessite une implication forte des métiers. Les données spécifiées seront autant que possible hiérarchisées (regroupement en famille) et les liens entre objets décrits.

> Remarque : on peut aussi profiter de cette analyse pour classifier les informations (ou inversement partir d'une opération de classification pour identifier les données de référence), connaître les propriétaires des données, l'application actuelle point de vérité ou son absence, les normes et conventions utilisées, les procédures…

Cette étape est fondamentale. Deux situations se présentent selon qu'il existe ou pas un plan d'urbanisme du SI pour la direction de l'entreprise au sein de laquelle le projet s'inscrit.

Dans le premier cas, il s'agit de croiser les objets et données du sous-système référentiel avec le périmètre fonctionnel du projet pour identifier le sous-ensemble des données de référence. Il est malgré tout nécessaire de s'assurer de l'exhaustivité des données de référence recensées en vérifiant la validité du plan. C'est en général l'occasion de procéder à sa mise à jour.

Dans le second cas, le plus simple est de partir des objets métier les plus structurants dans les processus et d'analyser les grandes fonctionnalités.

Une fois cette opération effectuée, il convient de procéder à la description des données nouvelles ou à la recherche des descriptions des données existantes et d'en vérifier la validité.

Il s'agit essentiellement de :

- Typer les données : cœur de métier (ou « maître »), données constitutives, données paramètre.
- Décomposer les données cœur de métier pour identifier et itérer les données constitutives et lister les données paramètre.
- **Identifier les propriétaires et les responsabilités, leurs exigences qualité** (fraîcheur, disponibilité, complétude…).
- Déterminer leur source : externe à l'entreprise, à la direction (SI d'une autre direction), ou interne au sous-système relatif au projet.
- Déterminer leur niveau de partage et d'échange dans le SI, voire dans l'entreprise : applications utilisatrices, modalités d'échanges…
- Identifier parmi celles-ci **une éventuelle application point de vérité.**

Au final, on classera les données de référence en trois groupes :

A. Les données existantes qui relèvent déjà de solutions de type MDM et/ou DQM : le projet doit alors se conformer aux règles et processus référentiels établis sauf arbitrage contraire.

B. Les autres données existantes : la mise en œuvre du projet doit être l'occasion de décider s'il y a lieu ou non de mettre en œuvre une solution du type précédent.

C. Les données nouvelles, *a priori* non encore partagées, pour lesquelles la direction doit statuer sur le type de solution.

Notons qu'un projet métier peut comporter les trois groupes alors qu'un projet de gestion de données de référence ne comportera au plus que les groupes B et C.

10.2.2 Identifier et décrire les processus référentiels

Rappel : un processus référentiel est un processus spécifique à la création, la modification ou la suppression d'une donnée de référence (cf. chapitre 3 Processus métier et processus référentiels).

Il convient donc de déterminer les processus référentiels de chaque donnée de référence identifiée. Dans un premier temps, le croisement de la donnée avec les processus métier qui relèvent du périmètre fonctionnel du projet permet d'identifier, pour chaque processus qui opère la donnée, sa composante métier et sa composante référentiel.

Dans un second temps, l'urbaniste ou l'architecte des données doit s'assurer que chaque donnée, quel que soit son groupe, possède bien au moins un processus de création, un processus de mise à jour et un processus de suppression.

Pour les données du groupe A, les processus référentiels sont déjà connus et opérationnels. Il faut néanmoins vérifier si on introduit ou pas de nouveaux cas ou variantes dans le projet qui nécessitent de faire évoluer les processus référentiels. Le projet implique également, et c'est sans doute là le plus important, d'établir la nécessaire coopération avec la solution MDM existante pour les processus à composante référentielle.

Pour les données du groupe B, selon qu'une solution de type MDM est choisie ou non, il s'agit de déterminer les processus référentiels en tenant compte en plus des processus mis en œuvre par les applications existantes.

Enfin, pour les données du groupe C, le projet a la garantie que son choix d'implémentation n'a aucun impact sur le SI existant. Si une solution de type MDM est retenue, l'entreprise ou la direction doit malgré tout inscrire dans son plan d'urbanisme le fait que les projets futurs qui opéreront ces données (du groupe A), devront s'appuyer sur la dite solution. Si aucune solution de type MDM n'est retenue, alors les projets futurs qui auront ces données se retrouveront dans la situation évoquée ci-dessus pour le groupe B. Là encore, l'entreprise ou la direction devra fixer les conditions de lancement d'un projet de type MDM. On notera que les entreprises « monarchiques » ou « fédérales » peuvent instituer plus facilement ces étapes de validation des projets et en confier la responsabilité au Comité de gouvernance, aux propriétaires et architectes de données.

On peut profiter de cette démarche processus référentiel pour décrire le cycle de vie métier des données concernées (qui peut être réalisé aussi dans les spécifications).

10.2.3 Spécifier les méthodes et règles de gouvernance

L'objectif est de produire un document qui spécifie les méthodes et règles de gouvernance retenues ou proposées et ce, pour chacune des données des trois groupes précédents. On parle ici de « gouvernance » mais aussi de sa déclinaison en processus de gestion appropriés aux données.

On pourra fortement s'inspirer de ce qui a été décrit dans le chapitre précédent et, s'il existe déjà des données du groupe A, partir des documents *a priori* existants et les confirmer, voire les amender.

10.2.4 Définir les principes de migration des données, analyser et assainir les données existantes

Dans cette tâche, nous incluons l'analyse et l'assainissement des données du SI existant, activités nécessaires pour la migration des données.

La migration de données désigne le processus de transfert de volumes, souvent importants, de données des sous-systèmes (ou applications) existants vers des sous-systèmes cibles. C'est le plus souvent une opération lourde, qui peut s'étaler sur plusieurs mois, voire plusieurs années (au fur et à mesure de la prise en compte des domaines au sein de la solution cible). Elle est donc coûteuse et souvent mésestimée. Et plus la période de migration est longue, plus il y a de chances que les sous-systèmes sources et cibles voient leur périmètre fonctionnel évoluer car « pendant les travaux, la vente continue ! ».

Les principales opérations à réaliser sont les suivantes :

- Analyser les données des systèmes sources. On peut s'appuyer utilement sur un outil de *profiling* DQM, surtout si les données sont dispersées sur plusieurs bases et/ou plusieurs sites. L'objectif est d'analyser la qualité des données existantes (cohérence, conformité, complétude, doublons…) et de constituer le modèle agrégé à partir des bases existantes.
- Assainir les données des systèmes sources. On peut s'appuyer sur l'analyse précédente, mais on peut aussi la compléter par un outil de nettoyage DQM capable de détecter des risques de doublons, de vérifier la conformité à un modèle (voir le chapitre 5). Certaines données doivent alors être corrigées manuellement et l'outil de nettoyage DQM vérifie que l'assainissement est bien réalisé.
- Établir les correspondances de modèles et transcodifications entre sources et cibles.
- S'assurer que les données migrées vont bien satisfaire aux éventuelles nouvelles règles de gestion et être conformes au nouveau cycle de vie, notamment en

cas de nouvelles règles, de durcissement des règles (par exemple un attribut qui était facultatif devient obligatoire) ou d'évolutions dans le cycle de vie des données (nouveaux états et transitions inexistants dans les systèmes sources).

- S'assurer de la cohérence des données de référence migrées à la fois dans le(s) sous-système(s) cible(s) métier et dans le sous-système référentiel (s'il existe).
- Tester et valider les procédures de migration.
- Migrer les données à l'aide d'un outil d'ETL.

La figure 10.3 schématise les principales opérations à réaliser lors d'une migration.

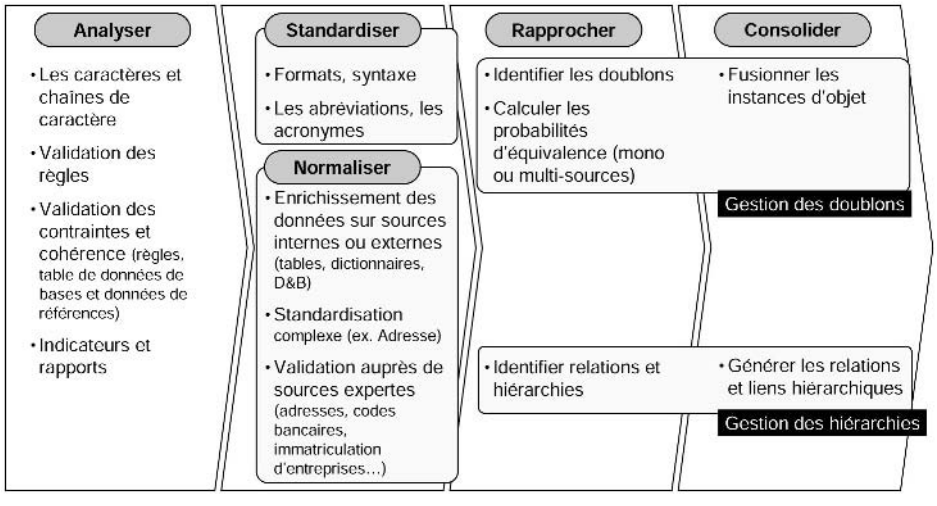

Figure 10.3 — Principales opérations à réaliser pour une migration de données

Concrètement, les auteurs ont pu valider tout l'intérêt d'outils de DQM lors d'un projet incluant la migration de données situées sur plusieurs bases de données et plusieurs sites (avec toutes les incohérences associées) vers une seule base cible. Le principe retenu a été de constituer progressivement une base intermédiaire avec toute la qualité requise (opération qui s'est étalée sur plusieurs mois) afin de migrer vers la cible à partir de ces données assainies, en minimisant les interventions sur les sources opérationnelles.

10.2.5 Définir les règles de qualité

Il s'agit de définir les règles de qualité à atteindre et à maintenir dans le système cible. On peut s'appuyer sur les résultats de l'analyse des données existantes effectuée avant la migration. Il s'agit d'une étape à ne pas négliger dans une démarche de gouvernance des données. La définition s'accompagnera d'indicateurs de suivi des règles.

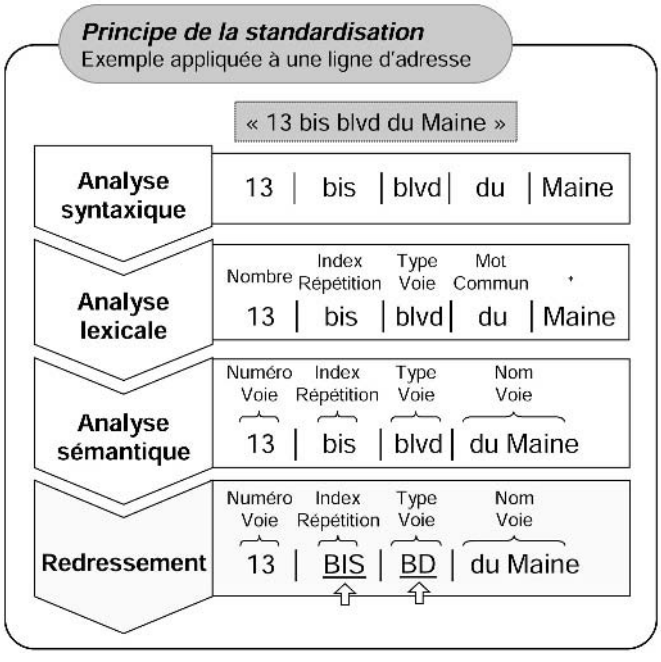

Figure 10.4 — Exemple de règle (standardisation)

10.2.6 Définir les contrats d'échange, de services

Cette tâche n'est pas spécifique aux projets de gestion des données de référence. Il est néanmoins vivement recommandé d'établir ce type de contrat pour tout échange de données entre sous-systèmes dès lors que le sous-système consommateur a lui-même des objectifs à respecter. Or c'est le cas dans les projets de gestion de données de référence qui mettent en œuvre une architecture de centralisation ou de coopération. Dans ces cas-là, les sous-systèmes de production sont dépendants des sous-systèmes référentiels.

Les contrats d'échange spécifient le niveau de service « contractuel » sur lequel s'accordent les parties concernées en fonction des exigences du consommateur et des contraintes du producteur :

- plage de disponibilité ;
- taux de disponibilité ;
- temps de réponse ;
- engagement de qualité…

Ce contrat établit *a minima* :

- Les rôles et responsabilités des acteurs de l'échange quant à son élaboration, son suivi et son évolution.

- Les spécificités de l'échange : données, qualité de service, fréquences, dates de mise à disposition, modalités de fourniture.
- Les procédures et l'organisation relatives au traitement des non-conformités et de leur qualification en anomalies avec niveau de gravité selon une échelle de référence, le délai de dépannage pour chaque niveau de gravité, l'instance d'arbitrage en cas de désaccord…
- Les procédures et l'organisation relatives aux évolutions fonctionnelles : fiche de demande, analyse d'impact, de faisabilité, estimation des charges, planification,
- Les procédures de traitement des dysfonctionnements, les modalités d'escalade, de reprise…

Le périmètre à prendre en compte tient compte de :

- la fréquence des échanges ;
- les interruptions de service et modes dégradés ;
- les rôles et responsabilités ;
- l'engagement des parties (fournisseurs et destinataires) ;
- la gestion et le suivi du contrat ;
- la gestion des anomalies et des modifications.

10.3 ÉLÉMENTS DE MÉTHODE

« Pensez grand, démarrez petit ! » Cet axiome est vrai aussi bien à l'échelle de la démarche globale (gouvernance et ensemble des projets référentiels) qu'à l'échelle d'un projet unique. Une vision à long terme ainsi qu'une approche itérative sont alors nécessaires.

Il est donc conseillé de lotir le projet en versions aisément maîtrisables. En effet la démarche est récente et doit être transverse aux différents domaines métier concernés :

- Les métiers ainsi que les techniciens doivent encore gagner en maturité.
- Les logiciels ne sont pas tous simples, ils peuvent être multiples (MDM plus DQM) et leur évolution génère versions et mises à jour.

Notre approche favorise les méthodes agiles (RUP, RAD, etc.[1]) et les méthodes de modélisation MDM ou SOA (MDM Alliance Groupe, Praxeme, etc.[2]). Des itéra-

1. Pour RUP, voir http://www.ibm.com/software/awdtools/rup/ et pour RAD http://www.entreprise-agile.com/
2. Voir la méthode de Pierre Bonnet du MDM Alliance Group : http://www.sustainableitarchitecture.com/mdm_alliance et le Praxeme Institute : http://dvauquier.free.fr/)

tions courtes avec une valeur métier rapide sont l'idéal, mais le contexte du MDM vous amènera peut-être à préférer d'autres méthodes.

Ainsi, dans le cadre d'une démarche mature, nous aurions préconisé des versions de dimension cohérente répondant à un rythme de deux à quatre versions par an. **Dans le cas contraire, nous préconisons une démarche différenciée avec des versions majeures répondant aux besoins métier et des versions mineures permettant l'alignement technique des solutions et la stabilisation fonctionnelle de la version en cours**. Nous proposons un enchaînement et une priorisation entre versions fonctionnelles et techniques pendant les mois nécessaires à l'acquisition de compétence des équipes. On prendra en compte les contraintes suivantes sur les versions pendant la phase de maturation :

- **Aspect fonctionnel** : définir des axes de progression sur lesquels chaque version majeure apporte une avancée. Nous proposons les axes suivants :
  – donnée : périmètre de données couvert par la version que ce soit en profondeur (nombre d'attributs) ou en largeur (type d'objet) ;
  – géographique : pays ou régions de déploiement du référentiel ;
  – métier : processus métier raccordés en amont (acquisition) ou en aval (consommation) du référentiel ;
  – fonction : types de services offerts par la solution.

- **Aspect technique** : définir et anticiper les apports d'une version de produit à un besoin métier. En début de maturation, décorréler les versions techniques des versions fonctionnelles. Une bonne vision des *roadmaps* des éditeurs est pour cela nécessaire. Participer aux clubs utilisateurs des éditeurs est un bon moyen d'être informé et même d'influencer sur le devenir du produit. On peut mener une étude d'opportunité de version en version afin de hiérarchiser leur priorité**.**

La figure 10.5 illustre cette mise en œuvre progressive.

Lancer son premier projet MDM comporte des risques et demande de se concentrer sur les gains en maturité et compétence des équipes. Afin de ne pas commettre d'erreur ayant des conséquences structurelles sur la solution, on peut suivre quelques-unes des recommandations suivantes.

- **Limitation du périmètre de données.**

Dans le cadre d'un projet monoréférentiel, on commence par les objets fortement partagés et les plus structurants. On peut nommer ce périmètre *« core data »* (ou cœur de donnée). Les **attributs éligibles** à ce périmètre sont en premier lieu les « **attributs discriminants** » (ceux qui permettent d'établir l'unicité d'un objet), viennent ensuite les « **attributs métier critiques** » (ceux qui ont été analysés comme les plus sensibles aux processus consommateurs) et enfin on n'oubliera pas quelques « **attributs de pilotage** » (qui permettent la mise en œuvre et le contrôle de la gouvernance, souvent plus techniques, et que l'on considère comme des métadonnées).

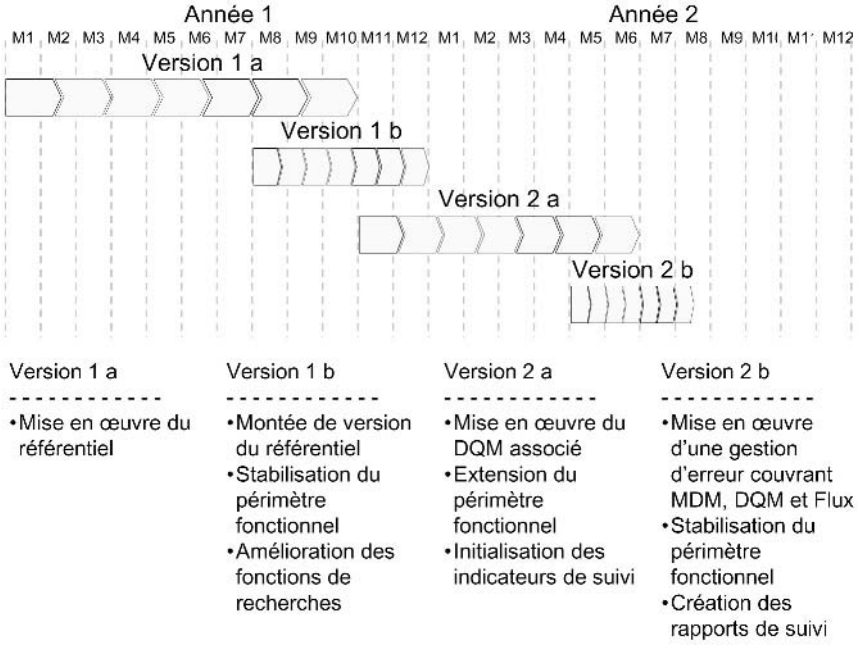

Figure 10.5 — Exemple de mise en œuvre progressive d'un référentiel

Dans le cadre d'un projet multiréférentiel ou monoréférentiel mais ouvrant sur des données constitutives majeures, soit un projet se définissant par de nombreux objet métier (client, produit, contrat, campagne, contacts, finance), on veillera à limiter le nombre de ces objets embarqués lors de la première version (2 à 3 objets maximum, répondant eux-mêmes à la logique *core data*).

Le *core data* est variable en taille en fonction de l'objet considéré mais 20 à 40 attributs sont les tailles fréquemment rencontrées.

La limitation du périmètre de données devrait aussi vous permettre de limiter le nombre des sources de données (et donc limiter l'effort de migration et d'intégration).

- **Limitation des interdépendances et contraintes**

La limitation du nombre d'objets métier favorise la limitation des contraintes de cohérence entre objets métier.

On limitera aussi la gestion des relations et des hiérarchies en embarquant les plus utiles si ce n'est les plus simples.

Pour les tables de contraintes (liste de valeurs), on utilisera de préférence un référentiel de paramètres. Mais si cette intégration supplémentaire engendre trop d'efforts, on peut inclure les contraintes directement à la solution et on diffère l'intégration au référentiel de paramètres.

- **Limitation de la complexité matricielle**

On peut limiter les étapes du cycle de vie pris en charge en limitant les profils utilisateurs, les spécificités géographiques…

- **Limitation des fonctions et services**

Ceci dépend de l'outillage utilisé (MDM, spécifique ou progiciel). Les outils MDM offrent le maximum de services et fonctions avec un minimum d'effort de mise en œuvre. On veillera cependant à s'aligner le plus possible sur les services et fonctions des outils.

10.4 CHARGES DE MISE EN ŒUVRE

L'analyse des charges de mise en œuvre d'un projet référentiel comporte des caractéristiques remarquables par rapport à la moyenne des projets :

- **Une charge de pilotage plus importante** à cause de la multiplicité des intervenants et des résistances associées (plus la donnée de référence est transverse et partagée, plus il y a d'intervenants et de freins).
- **Une charge de spécification plus forte** de par la nature transverse de la solution, le nombre potentiel de couches applicatives concernées et la prise en compte des différents processus référentiels.
- **Une charge de tests** (plan, jeux, et déroulement) plus importante du fait des interactions et différents points de tests nécessaires (aspect matriciel des spécifications) mais aussi de la nécessité d'automatiser ces tests (non-régression en production).
- **Une charge d'intégration** dépendant directement du mode de déploiement du référentiel (« *Big Bang* », par périmètre ou avec préservation du *Legacy*).

Nous avons exclu l'architecture « Répertoire virtuel » de cette analyse car elle n'offre que peu de points communs, que ce soit au niveau de la méthode ou de la réalisation des solutions.

Pour les autres architectures, l'effort de développement augmente en fonction de leur complexité, de la moins complexe à la plus complexe :

- 1. Consolidation.
- 2. Centralisation.
- 3. Coopération.

La figure 10.6 évalue l'effort de mise en œuvre en fonction de l'architecture.

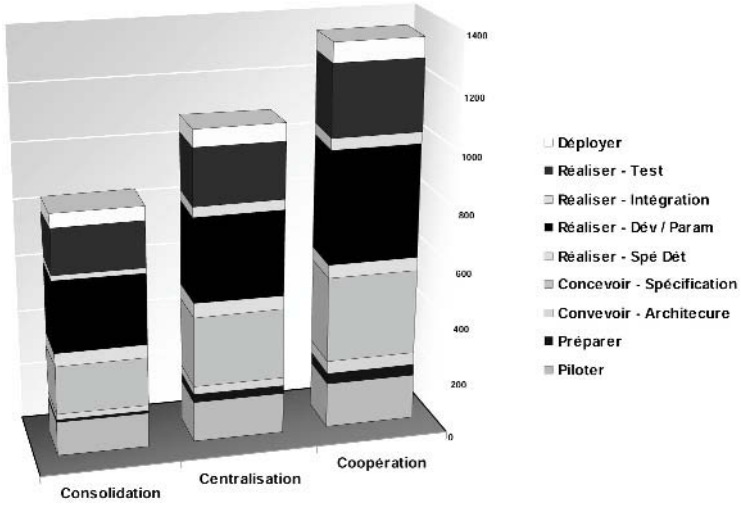

Figure 10.6 — Effort de mise en œuvre en fonction de l'architecture

10.5 BONNES PRATIQUES

10.5.1 Méthodes

Le tableau 10.1 liste quelques bonnes pratiques en termes de méthode.

Tableau 10.1 — Bonnes pratiques en termes de méthode

Bonnes pratiques en termes de méthode	Priorité */**/***
Impliquer les métiers dans les projets de gestion des données de référence.	***
Donner la priorité de méthode au **prototypage**, à **l'itération** (gestion de versions, en limitant les données prises en compte dans la première version) et au retour d'expérience.	***
Recenser et tenir à jour les **normes, standards** et **conventions** utilisables (nommage, classification, type, vocabulaire…). S'appuyer sur les normes, les standards et les conventions.	***
Identifier, spécifier et contrôler la conformité du projet aux **exigences réglementaires** (quelles exigences pour quelles données) et à la sécurité.	***
Ne pas sous-estimer la phase **d'analyse** et de **conception** dans un projet de gestion des données de référence (plus importante que dans un projet classique).	**

Bonnes pratiques en termes de méthode	Priorité */**/***
Capitaliser les informations et communiquer vers les projets et les métiers au sujet des activités liées aux données de référence. **Réutiliser** dans la mesure du possible ce qui existe pour les données de référence dans le cadre d'un projet, effectuer un **retour d'expérience** du projet.	**
Assurer la **dimension qualité** des données de référence dans les projets (qualité et fiabilité des systèmes sources, niveaux de qualité à atteindre pour les données de référence…).	**
Identifier les **contrats d'échange** des données de référence (disponibilité, qualité…) à mettre en place pour les données échangées entre organisations.	**

10.5.2 Organisation

Le tableau 10.2 liste quelques bonnes pratiques en termes d'organisation.

Tableau 10.2 — Bonnes pratiques en termes d'organisation

Bonnes pratiques en termes d'organisation	Priorité */**/***
Gérer (identifier ou attribuer) les **rôles et responsabilités** pour les données de référence (cf. rôles définis dans le chapitre sur la gouvernance).	***
Identifier et mettre en place les **procédures de gestion** des données de référence (création, modification, validation…) et **les droits associés.**	***
Établir les **règles de gouvernance des données de référence au niveau Entreprise** : – Privilégier la logique transverse à l'entreprise par rapport à la logique projet. – Définir les organisations liées à la gouvernance (comité de gouvernance…). – Mettre en place l'organisation nécessaire (transverse et par domaine).	***

En résumé

Un projet de gestion des données de référence induit des tâches spécifiques (identifier les données de référence, décrire les méthodes et règles de gouvernance, spécifier comment assainir et faire migrer les données…). Bien entendu, commencer par **identifier les données de référence** est un préalable. Le partage au sein des applications du SI est un critère déterminant. Si une migration des données est nécessaire, on en profitera pour **analyser la qualité des données existantes et constituer un référentiel de qualité satisfaisante** pour la migration. Cela peut nécessiter la mise en œuvre d'outils de DQM. Au-delà des outils, quelques **bonnes pratiques** s'imposent, comme le prototypage, l'itération, l'implication des métiers, le respect de normes et standards, la spécification de rôles et responsabilités…

11

Points clés à retenir

Objectif

Ce chapitre a pour objet de rappeler au lecteur les facteurs clés de succès dans la mise en œuvre d'une gestion de données de référence. Il convient de rappeler quelques règles de base, les bonnes pratiques essentielles, l'intérêt d'une démarche progressive et d'une vision prospective.

11.1 RÈGLES DE BASE

Elles sont au nombre de cinq. La figure 11.1 résume ces règles.

Identifier et décrire les données de référence

Cette étape est bien entendu indispensable et primordiale. Ce sont les **objets métier** manipulés qui servent à cette identification. Ils induisent les données « maîtres » à partir desquelles on peut décliner les données « constitutives » et « paramètres ». **Un des critères essentiels est aussi le partage entre plusieurs applications**. Rappelons qu'un paradigme de référence n'est pas relatif à une activité unique. Par exemple, le contrat étant une donnée opérationnelle pour la vente, il devient une donnée de référence pour les activités de l'entreprise en aval de la vente, tels l'administration des ventes, le recouvrement, la gestion des comptes… Mais le contrat est aussi une donnée de l'entreprise amont pour les achats. Il décrit alors la relation entre l'entreprise et ses prestataires ou partenaires. On identifie régulièrement ici deux natures de contrats. On y répond généralement avec deux référentiels, sauf si l'entreprise possède un fort recouvrement entre clients et fournisseurs.

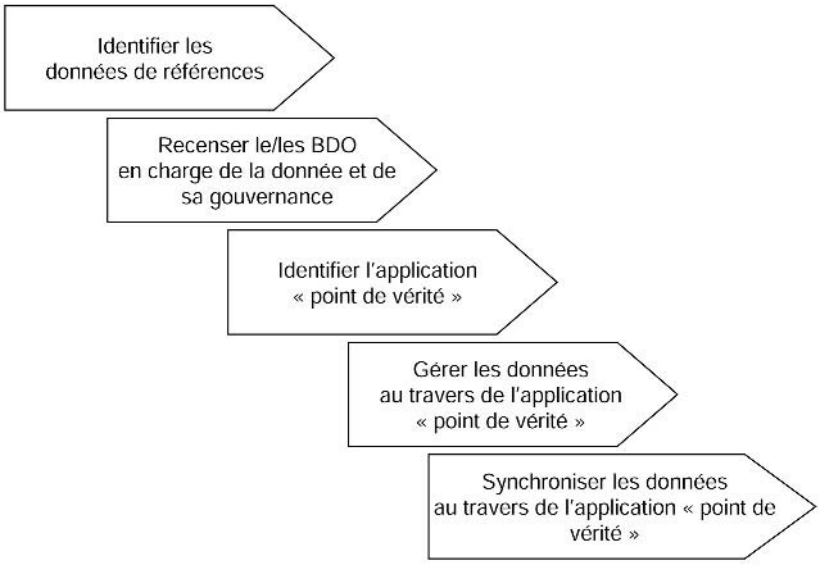

Figure 11.1 — Règles de base de la gestion des données de référence

De manière plus générale, il est essentiel de distinguer dans un SI les données de référence des données opérationnelles et décisionnelles. Par exemple, dans un SI qui traite de services aux clients d'un fournisseur d'éléctricité, les informations sur les locaux d'un client sont des données de référence dont ce SI est le garant (voir paragraphe sur la notion de propriétaire) alors que les données relatives aux services sont des données opérationnelles. Les données de référence identifiées doivent être l'objet d'une attention plus particulière (en termes de qualité notamment) car elles sont susceptibles d'être utilisées par plusieurs applications. Les données opérationnelles ou décisionnelles sont, elles, davantage liées à une application précise. Au niveau de l'implémentation, il faut au moins prévoir des tables différentes, voire des SGBD distincts.

Recenser le (ou les) propriétaires en charge de la donnée et de sa gouvernance

Les règles de gestion des modèles et des instances sont susceptibles d'évoluer, d'où la nécessité d'avoir identifié ou désigné un « **propriétaire** » capable de valider leur évolution. Si cette opération n'est pas indispensable pour la totalité des données de référence, *a contrario*, elle l'est pour les données « maître » et s'accompagne d'une mise sous gouvernance.

Il n'est pas rare d'utiliser un modèle livré avec un progiciel. Cela ne dispense en aucun cas de mener à bien cette tâche d'identification d'un propriétaire car il est indispensable de déterminer le périmètre des attributs utiles au projet parmi tous les attributs proposés par le progiciel. Il faut aussi indiquer comment ces attributs sont mis en œuvre concrètement par rapport aux besoins métier. De plus, il reste indispensable de définir les règles de sécurité et de qualité.

Rappelons la possibilité déjà évoquée de l'existence de plusieurs propriétaires d'un modèle, chaque propriétaire étant responsable d'un certain nombre d'attributs. L'organisation doit désigner dans ce cas un propriétaire principal (*business data owner*) ou une instance d'arbitrage (sorte de « syndic »).

Il est important de noter également que le modèle retenu n'impose pas l'ensemble de ses attributs aux autres applications. Elles peuvent n'utiliser qu'un sous-ensemble des attributs définis (à gérer lors de la diffusion) ou posséder des attributs complémentaires gérés localement.

Que faire lorsque deux ou plusieurs progiciels imposent des modèles divergents ? Il faut arriver à définir ce qui est commun et choisir l'un des progiciels comme point de vérité ou bien faire gérer les référentiels hors de ces progiciels.

Cette notion de propriétaire est en fait très liée à celle de point de vérité rappelée dans le paragraphe qui suit car le propriétaire détient de fait ce point de vérité. Si nous reprenons l'exemple d'un SI services qui gère les données relatives aux locaux d'un client, le propriétaire fait nécessairement partie du SI services (au moins pour les attributs validés par le SI services).

De manière plus générale, il faut mettre en place la gouvernance. En plus de l'identification des propriétaires, cela suppose de :

- Formaliser l'organisation. Il n'existe pas de règles *a priori*, néanmoins, nous avons vu tout l'intérêt du rôle **d'architecte des données**. Il est en particulier en charge de la capitalisation et de la cohérence des modèles et des flux de données.
- S'attacher à la définition de critères de qualité et au **suivi de cette qualité.**
- Définir les règles et procédures de gestion.

Identifier l'application « point de vérité »

C'est le minimum absolument nécessaire. Il faut en effet être capable **de repérer l'application à partir de laquelle les données de référence sont considérées comme valides et donc diffusables auprès d'autres applications**. Plus les données de référence (constitutives et paramètres) sont fines, plus nombreuses sont les applications candidates et plus un arbitrage niveau gouvernance s'avère nécessaire. Dans ce cas-là, il peut sembler nécessaire de mener un projet afin de créer cette application point de vérité.

En toute logique, cette application point de vérité ne devrait pas être une application décisionnelle qui se situe en bout de chaîne dans un SI. En effet, le but est d'assurer un partage des données de qualité entre un maximum d'applications. Et se limiter aux applications décisionnelles est assurément un constat d'échec lorsqu'on a réconcilié des données produites et diffusées sans la cohérence et la qualité requises.

Cette notion de point de vérité peut aussi être considérée par rapport à un ensemble d'attributs. Il est possible d'identifier plusieurs points de vérité pour une même donnée. Ainsi, certains attributs d'un local peuvent être gérés au sein d'un CRM et certains attributs complémentaires l'être au sein d'une application qui pro-

duit un service pour un client. Le CRM est point de vérité pour les applications qui n'ont besoin que d'un nombre réduit d'attributs ; l'application service l'est pour les applications qui nécessitent l'ensemble des attributs. Remarquons toutefois que, dans ce cas, on pourrait aussi considérer que l'application service est le seul point de vérité, le CRM participant à l'acquisition de la donnée. On voit que ces notions se révèlent étroitement liées à la gouvernance.

Toute application « point de vérité » d'une donnée doit permettre sa diffusion la plus large possible et son interrogation à travers certains outils de requête ou les services *ad hoc*.

Néanmoins, se contenter d'identifier les applications points de vérité peut se révéler insuffisant.

- Cela oblige à gérer de manière explicite les données de référence avec le niveau de qualité requis dans des applications opérationnelles dont ce n'est pas la finalité.
- Cela contraint à mettre en œuvre les interfaces d'échange nécessaires à la diffusion vers les applications utilisatrices desdites données.
- Le risque est fort d'adopter un modèle de données imposé par un progiciel peu évolutif dans le temps et dont les évolutions sont, elles aussi, imposées par l'éditeur.
- Cela empêche souvent d'avoir une vision unique d'un ensemble d'objets métier à travers une seule application.
- C'est parfois inapplicable s'il est difficile d'identifier un point de vérité unique (application qui a une vue incomplète d'un objet, à travers quelques attributs, alors que d'autres applications sont points de vérité pour d'autres attributs). C'est également le cas pour les données constitutives (type adresse) et tables de paramètres (type liste des communes, codes postaux) que l'on retrouve dans plusieurs applications du SI, voire différents SI.

Il faut alors examiner la possibilité de gérer les données de référence dans une application séparée de type MDM. La **gestion des données de référence est ainsi indépendante des applications opérationnelles**. Toutefois, ce n'est pas une règle absolue : un progiciel métier peut être « point de vérité » ! **Cela n'oblige pas nécessairement à séparer les données des traitements dans les applications** (copie possible des données à partir de l'application point de vérité). **Cela permet de séparer gestion et diffusion des données de référence des applications opérationnelles non conçues pour cette gestion et ces échanges.**

Gérer les données de référence à partir ou au travers de l'application point de vérité

Comme mentionné, on favorise les solutions placées au plus haut dans la chaîne de l'information. Les architectures de centralisation ou de coopération ont notre préférence. Elles induisent tout ou partie des fonctions de gestion suivantes :

- Définir des modèles.
- Saisir ou importer des instances.
- Modifier des modèles.
- Modifier des instances.
- Journaliser les créations, modifications…
- Historiser les données et métadonnées.
- Versionner les référentiels.
- Les sauvegarder.
- Gérer un *workflow* de saisie si nécessaire.
- Gérer la qualité.
- Gérer des dates d'effet (date à partir de laquelle les nouvelles données doivent être prises en compte…).
- Supporter des règles de diffusion des données validées (voir la section qui suit).

Gérer la synchronisation des données de référence en relation avec l'application point de vérité

Gérer la synchronisation des données de référence, c'est se doter d'une vue d'ensemble de la répartition de la donnée et de ses flux à l'échelle du SI.

Dans le cadre de la mise en œuvre d'un référentiel, une telle vision d'ensemble repose, en premier lieu, sur la diffusion de la donnée. Cela consiste en plusieurs opérations possibles :

- Pilotage de la diffusion (quel événement provoque la diffusion, selon quel moyen d'échange…).
- Diffusion sélective d'attributs.
- Prise en compte d'une date d'effet.
- Suivi de la diffusion.

Cela signifie aussi que la mise à jour d'un paradigme est déclenchée par un ou n processus référentiel. Ainsi, l'événement déclencheur de la diffusion de l'information est plus dépendant du « point d'acquisition » de l'information que du « point de vérité ». Pour simplifier les échanges, chaque attribut partagé d'un paradigme doit à terme être « sourcé » au sein du point de vérité. Cela signifie que **l'éligibilité de la donnée**, l'analyse des **flux**, des **événements déclencheurs** de la synchronisation et la mise en œuvre d'un **plan de transformation** doivent viser l'agrégation de tous les attributs partagés ainsi que leur gestion de synchronisation depuis le référentiel. Ce n'est pas réalisable dès la première version de la mise en œuvre d'une solution mais cela doit être pris en compte lors de la création d'une initiative et de sa trajectoire (*roadmap*).

11.2 BONNES PRATIQUES ESSENTIELLES

La figure 11.2 synthétise les bonnes pratiques essentielles.

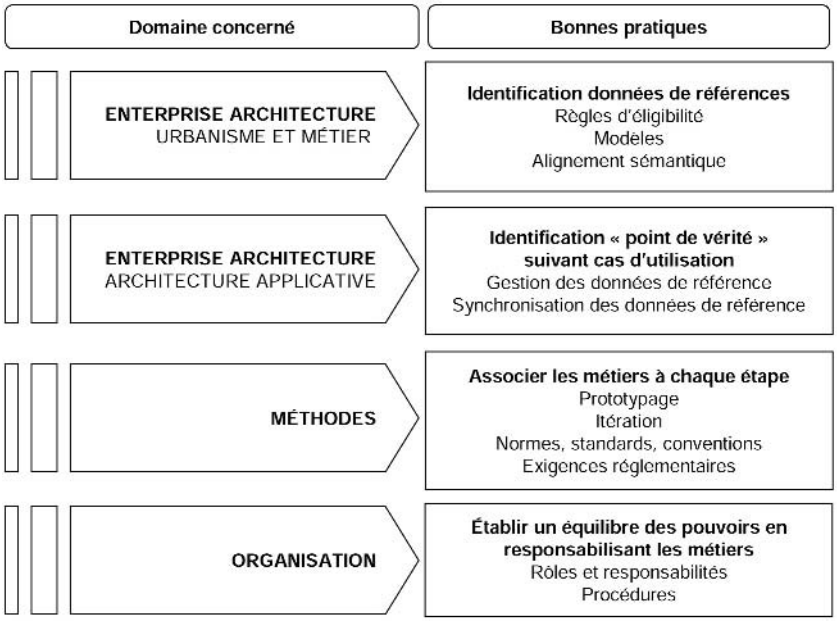

Figure 11.2 — Bonnes pratiques essentielles

11.3 VISION PROSPECTIVE ET PROGRESSIVE

Souvent aggravée par la multiplication et l'accélération des fusions-acquisitions, la fragmentation des données dans des systèmes hétérogènes séparés pousse **de plus en plus d'entreprises à opter pour une gestion centralisée de leurs données de référence**. Cela consiste à regrouper et gérer dans un référentiel unique tout ou partie des données de référence de l'entreprise à l'aide d'une solution MDM du marché, d'un progiciel ou grâce à un développement spécifique.

Quelle que soit la solution retenue, l'entreprise doit l'intégrer avec ses différents systèmes métier. C'est une **approche qui tend vers une plate-forme d'intégration des données d'entreprise**. Celle-ci fournit toutes les fonctionnalités nécessaires pour accéder, intégrer, migrer et consolider les données de référence. Cela contribue à réduire la complexité, garantir la cohérence et dynamiser l'entreprise.

Cette approche est nécessairement **transverse**. En première analyse, elle s'apparente à la mise en place de solutions d'échanges inter-applications (EAI, ESB, annuaire de services…).

Mais la démarche est rendue complexe par une architecture applicative plus étendue, par le nécessaire alignement des processus et de la sémantique et par l'accompagnement métier et organisationnel rendus indispensables.

Il est donc essentiel :

- D'isoler les données de **manière progressive** (sauf dans le cas de la refonte d'un SI) ;
- D'avancer vers **une cible** (urbanisme et gouvernance).

Cela signifie, au périmètre d'un projet, de :

- définir un périmètre restreint mais offrant un gain sensible afin de faciliter l'adhésion des acteurs. Par exemple, débuter un projet MDM pour centraliser des données externes ou gérer les principales données paramètres peut être une bonne stratégie ;
- lotir afin d'assurer la courbe d'apprentissage.

Cela implique, au périmètre du SI, de :

- Penser et coordonner chaque projet en accord avec une **vision d'ensemble au travers d'une politique et d'instances de gouvernance**.
- Mettre en place une infrastructure mutualisée et industrialisée.

En résumé

Nous avons dans ce chapitre rappelé les **règles de base** : identifier et décrire les données de référence, recenser le ou les propriétaires, identifier ou définir les applications point de vérité, gérer et synchroniser l'ensemble à partir de ces applications point de vérité. Nous avons également récapitulé quelques **pratiques essentielles**, dont l'implication indispensable des métiers et de la gouvernance (rôles et responsabilités, niveau de qualité à spécifier, atteindre et maintenir).

Conclusion

La masse des données ne cessant de croître, l'enjeu de la gestion des données est essentiel. Tout au long de cet ouvrage, nous nous sommes efforcés d'exposer comment améliorer la gestion des données. Nous avons analysé les technologies, présenté des pratiques pertinentes et des solutions efficaces et, plus particulièrement à nos yeux, le MDM. Nous avons insisté sur l'importance de la gouvernance et la spécificité de la gestion des données dans les projets.

Nous concluons en nous interrogeant sur les perspectives dans le domaine de la gestion des données et sur les implications au sein du SI. Progresse-t-on :

- vers une architecture de l'information ?
- vers un nouveau système d'information ?

Vers une architecture de l'information ?

Avec l'évolution des architectures orientées services, une attention plus grande doit être portée à la notion d'architecture des données. Mais si les initiatives SOA commencent à être comprises, de nouvelles opportunités apparaissent pour transformer une vision technique et normative des données en une vision métier.

Les architectures SOA dissocient les données des processus et les applications de leurs interfaces**. L'un des facteurs clés du succès de ces architectures est de savoir où se situent les données, comment s'y connecter (plate-forme d'intégration de données) et où stocker les données de référence (MDM).** Elles rendent nécessaire une solution d'administration des données d'entreprise. C'est une nouvelle couche transverse à tous les référentiels mais aussi, demain, aux données transactionnelles pouvant être accédées par les applications métier orientées services. Selon une étude Gartner, c'est l'une des technologies clés de l'avenir.

L'entreprise a vu ces dernières années une accélération du changement de son écosystème. Elle doit faire face à des défis économiques de plus en plus grands : la volatilité des offres sectorielles, la dérégulation, l'ouverture du marché et la plus grande agressivité de ses acteurs. **La flexibilité (grâce à la SOA) et le partage de l'information sont une réponse à ces défis** mais il est nécessaire de s'interroger : qu'en est-il de la qualité des données et de la pertinence de l'information qui font référence pour les acteurs de l'entreprise ?

Les données de référence de l'entreprise, par nature partagées, sont la base indispensable dans la capitalisation de son savoir-faire.

Les solutions MDM sont aujourd'hui une solution possible au cœur des données de référence. Elles supposent une vision qui prenne en compte la **gouvernance** des données de référence. Pour que chaque corps de métier accède à la donnée de référence selon ses propres besoins, au moment opportun, il faut une infrastructure qui facilite la diffusion de ces données.

La gestion harmonieuse et maîtrisée des données de référence permet la mise en place de processus de gestion et de systèmes d'analyse fiables et homogènes qui offrent à toutes les personnes impliquées un accès aux mêmes connaissances et informations. C'est pourquoi une solution permettant à la fois la consolidation des données de base et l'accès à des données globalement cohérentes sur tous les systèmes confère un avantage concurrentiel décisif. Grâce à cette méthode, la qualité et la cohérence des données sont préservées via les mécanismes de contrôle et de validation.

La figure conclusion.1 schématise les principales recommandations que nous avons énoncées.

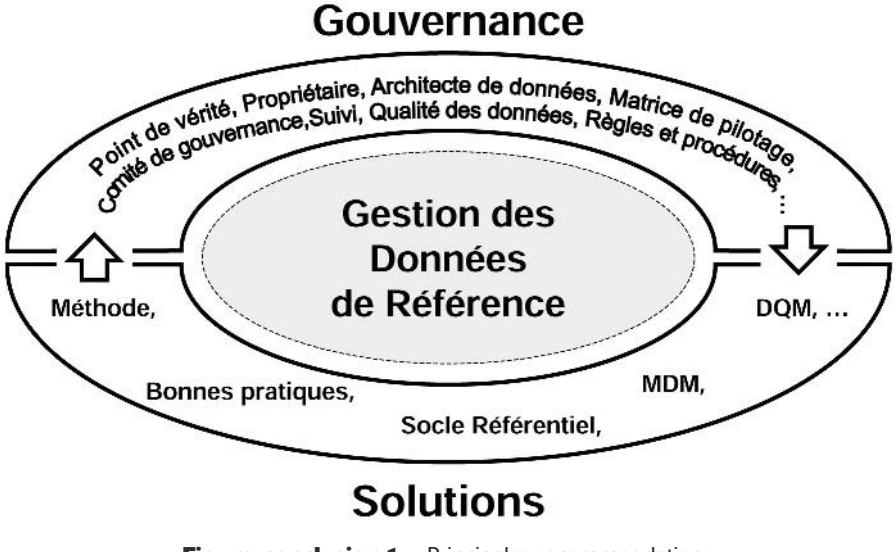

Figure conclusion.1 – Principales recommandations

Amélioration des stratégies EIM

Cependant, si la qualité des données est une étape essentielle en ce qu'elle structure le système d'information, elle n'offre pas encore toutes les promesses de l'EIM (*Enterprise Information Management*). La donnée n'est pas encore information. La structuration du SI par des données fiables et contrôlées est un premier pas, par exemple pour le décisionnel, vers l'obtention d'une information sûre. **Mais l'information est affaire de subjectivité et elle est aussi contenue au sein de données non structurées** (documents, éléments multimédias…).

La mise en place d'une architecture orientée services et de référentiels de données comme support des processus métier de l'entreprise offre à terme une couche de données ainsi que les fonctionnalités nécessaires à une stratégie de gestion des informations au niveau de l'entreprise (EIM).

Cette évolution passe par un double mouvement :

- l'extension des notions de gouvernance des données structurées aux données non structurées ;
- une qualité de l'information pressentie non seulement pour ses valeurs intrinsèques mais, de plus en plus, pour son adéquation au contexte et à l'utilisateur.

On assiste à la convergence des fonctionnalités des solutions de gestion de contenu non structuré (ECM, *Enterprise Content Management*) et de gestion des données structurées dans une architecture orientée service. La mutualisation des pratiques offre une vision du capital d'informations de l'entreprise. Au sein de notre discipline, cette évolution est aussi sensible du fait de la part grandissante que prennent les fournisseurs d'information tiers (Dun & Bradstreet, Fininfo…).

Notamment, l'accent mis sur l'enrichissement sémantique des données de référence apporte une forte valeur ajoutée métier. Disposant ainsi de données de référence métier de qualité et disponibles, les solutions MDM gèrent les liens conceptuels structurants entre ces données :

- enrichissement du descriptif des données (contenu métier ou technique) ;
- mise en relation sémantique des données sous forme de structures organisées (hiérarchies de sens, collections de données, ontologies : classifications, taxinomies, thésaurus).

Le MDM sémantique constitue ainsi la première brique d'une stratégie d'entreprise de gestion de l'information autour d'un socle EIM.

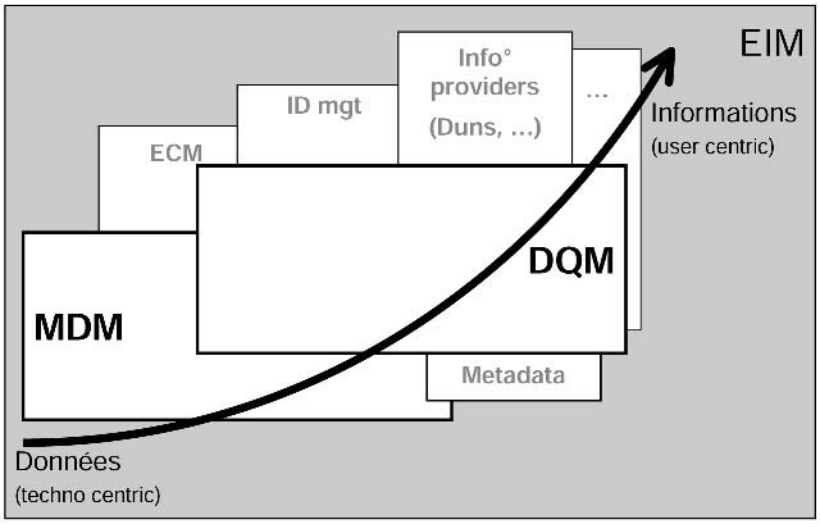

Figure conclusion.2 – Évolution vers l'EIM

Vers une gestion du capital d'informations de l'entreprise

Peu d'entreprises adoptent encore une stratégie de gestion centralisée du **capital d'information** au niveau groupe par la mutualisation des initiatives de l'EIM (MDM, DQM, ECM…) mais il est intéressant de noter l'émergence de certains standards : RDF (*Resource Description Framework*), spécification du W3C permettant de traiter des métadonnées, OWL, *Ontology Web Language*, langage développé par le W3C pour une meilleure interprétation des contenus Web par les applications devant traiter ces données sans pour autant devoir les présenter à des êtres humains. Ces standards laissant entrevoir des possibilités d'analyse sémantique à forte valeur ajoutée métier sur ce capital d'informations. Par exemple, les ontologies donnent la possibilité de relier conceptuellement (sémantiquement) les ressources en vue de leur exploitation métier ou technique.

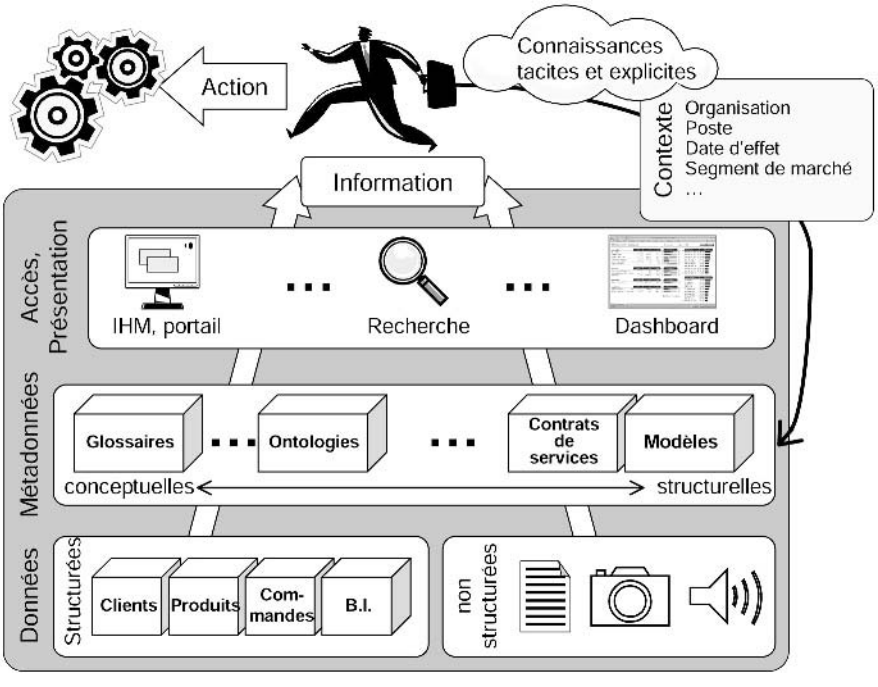

Figure conclusion.3 – Vue conceptuelle de la gestion des informations en entreprise

Un tel accès aux données (structurées et non structurées) au travers de filtres contextuels (de niveau entreprise, puis domaine métier puis spécifiques à chaque utilisateur) préfigure la gouvernance de l'information et les SI de demain. Mais aujourd'hui ne mettons-nous pas déjà en place de nouveaux systèmes d'information ?

Vers un nouveau système d'information ?

La gestion efficace des données, et plus particulièrement des données de référence, passe par une prise de conscience et une démarche volontariste. C'est à ce prix qu'émergera un nouveau système d'information (généralisation ici de la notion de sous-système référentiel évoquée au chapitre 3). Nous avons introduit dans cet ouvrage la notion de référentiel, qui permet de gérer de manière autonome les données de référence afin de les distribuer ensuite aux autres applications du SI. Bien entendu, il s'agit d'une vision cible, mais on peut la mettre en œuvre progressivement dans le cadre d'une démarche opportuniste, c'est-à-dire à l'occasion de projets métier.

La figure conclusion.4 illustre ce SI cible idéal. C'est une vision logique (d'autant plus que dans une approche SOA peuvent exister des interactions entre SI ou sous-systèmes) : le SI référentiel n'est pas nécessairement physiquement séparé du SI opérationnel.

L'essentiel est en fait de garantir trois fonctions :
– disposer d'une référence reconnue (point de vérité) ;
– faciliter la gestion (modèles et hiérarchies de modèles, relations entre modèles, qualité, historisation, sécurité…) ;
– permettre les échanges avec les applications consommatrices afin d'assurer la cohérence des données.

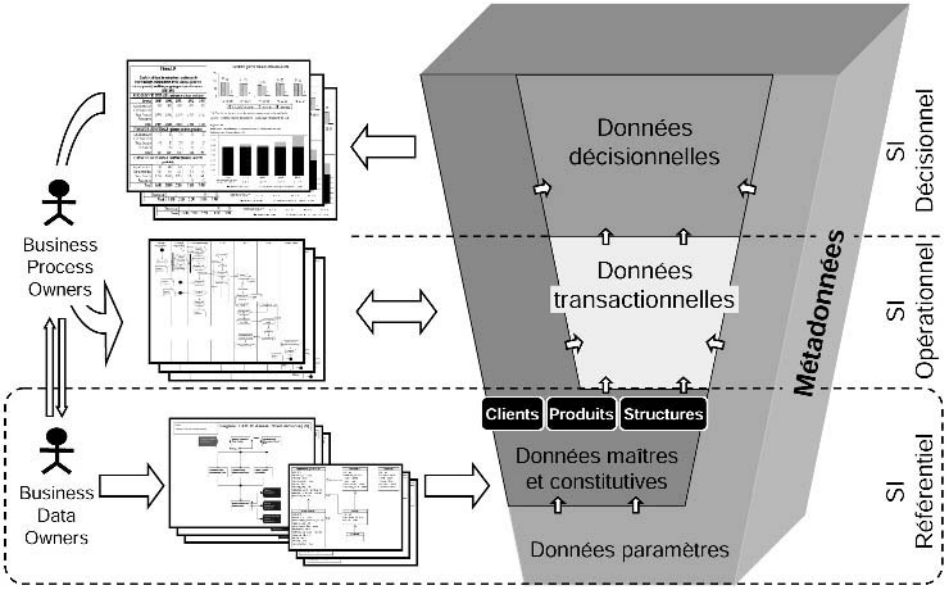

Figure conclusion.4 – SI cible

L'interaction entre les processus opérationnels et l'administration des données de référence au sein d'une même application peut poser des problèmes de responsabilité et de sécurité (qui peut faire quoi, avec quels droits ?) mais peut aussi entraîner des données de qualité moindre (les processus opérationnels étant prioritaires). De manière plus générale, décorréler les données des processus augmente la flexibilité du système d'information.

Cette stratégie vise à augmenter l'indépendance du système d'information face aux progiciels. L'objectif est d'unifier la gestion des données de référence. Les progiciels manipulent des données de référence, souvent en grand nombre car elles débordent sur les stratégies de paramétrage fonctionnel et technique. Les entreprises peuvent constater, suivant leur contexte, les contraintes suivantes :

- Le progiciel dispose de fonctions de type MDM mais leur usage n'est pas suffisamment ergonomique pour les équipes métier. Les fonctions de gouvernance à forte valeur ajoutée comme la gestion des versions, la gestion des droits, la valorisation par contexte, l'auditabilité… ne sont utilisables que par les informaticiens ou n'existent pas.

- Les fonctions de type MDM, intégrées au progiciel, sont utilisables uniquement dans son périmètre. Dans une architecture multiprogiciel, les utilisateurs n'ont pas d'outil d'administration unifié des données et la cohérence entre les référentiels détenus par différents progiciels est difficile à gérer.
- Les développements spécifiques ont aussi leurs propres solutions de type MDM qui s'ajoutent à celles des progiciels, ce qui augmente la création de silos fonctionnels et techniques de gestion des données de référence.

Parmi les solutions possibles pour construire ce SI référentiel, les progiciels de MDM permettent une mise en œuvre facilitée de ces trois fonctions. Selon le Gartner, d'ici 2010, 70 % des entreprises devraient avoir inscrit le MDM parmi leurs priorités.

Ces produits MDM présentent un intérêt notable si au moins deux critères parmi les quatre suivants sont présents :

- nombre important d'applications consommatrices des données de référence ;
- données assez peu figées dans le temps ;
- attributs et règles de gestion non implémentables simplement dans une application opérationnelle de type progiciel ;
- évolution ou refonte du SI pour une meilleure agilité (le MDM est une des briques possibles des architectures SOA).

> Il est essentiel de réaliser un POC (*Proof Of Concept*) et une étude de retour sur investissement avant de se lancer dans un projet MDM, sachant qu'un tel projet doit reposer sur la mise en œuvre de règles de gouvernance et sur un accompagnement fort du métier.

Rappelons, car c'est fondamental, **que le MDM est une brique transverse entre applications qui fait partie des briques mutualisables d'un système d'information, au même titre qu'un EAI, un ESB ou un annuaire de services**.

> Les évolutions importantes du SI liées à des changements d'organisation ou à de nouvelles contraintes techniques ou réglementaires sont particulièrement favorables à ce type d'approche.

De manière plus générale, le MDM est un facteur de plasticité du SI, puisque les modèles et instances y sont gérés et plus simplement adaptables que si l'on utilise un progiciel ou un développement spécifique pour implémenter un référentiel. Enfin seule cette approche permet de garantir des données consultables, à jour et de qualité, et non des données dispersées dans des silos applicatifs.

Arriver à mieux gérer les données est encore un réel enjeu, et ce pour toutes les entreprises. La révolution en cours consiste à passer d'une gestion « centrée sur les traitements » à une **gestion « centrée sur les données ».**

Il est désormais temps de s'attaquer à ce problème de fond et nous espérons y avoir contribué en présentant à la fois les méthodes et les outils nécessaires. **Les entreprises qui, les premières, sauront mieux gérer leurs données disposeront d'un avantage compétitif indéniable.**

Annexes

A

Modélisation des données

Annexe A – Modélisation des

Objectif

Au-delà des rappels méthodologiques sur la modélisation des données, cette annexe a pour but d'apporter des réponses concrètes à un domaine de la modélisation sous-estimé en général par les entreprises : la modélisation des flux de données en vue de leur industrialisation**.**

Un **modèle de données** est une représentation graphique simplifiée de la réalité. Il est composé d'un ensemble d'entités, d'associations, de propriétés et de contraintes et rend compte d'un sujet donné. Ce sujet peut concerner aussi bien un domaine fonctionnel, une activité métier, un objet métier, une organisation, un flux d'information, un document… enfin tout sujet d'intérêt pour l'entreprise.

Nous nous intéresserons donc ici à la fois aux **modèles de bases de données relationnelles et aux modèles de document échangés** entre sous-systèmes. Ils sont respectivement destinés à définir la structure de stockage des données dans des bases et la structure des données échangées à travers des **flux**.

A.1 OUTILS POUR CRÉER UN MODÈLE DE DONNÉES

Alors que les modèles simples de données (ceux qui consistent en un faible nombre de tables ou d'objets) peuvent être créés « manuellement », les modèles plus complexes nécessitent une approche plus systématique.

Il existe deux grandes communautés dans le domaine :

- La communauté MERISE qui utilise la méthode éponyme s'appuyant sur le modèle entité-relation, qui est très largement utilisée dans le domaine des bases de données relationnelles et pour laquelle nombre d'outils existent sur le marché.
- La communauté de la programmation orientée objet, qui utilise le langage UML (*Unified Modeling Language*) s'appuyant sur un modèle plus puissant que le précédent : le modèle objet qui permet notamment de modéliser à la fois les données et les traitements. Dans UML, les « diagrammes de classe » présentent beaucoup de ressemblances avec les diagrammes entité-relation mais la plupart des outils UML, comme Rational Rose et Embarcadero Describe, n'ont pas la possibilité de supporter la traçabilité entre les niveaux conceptuel, logique et physique.

Nous rappelons ici les concepts principaux utilisés par MERISE et UML et renvoyons le lecteur à la littérature spécialisée et aux multiples sites Web pour une description détaillée.

A.2 MODÉLISATION MERISE

A.2.1 Modèle conceptuel de données (MCD)

Une **entité** (ou **individu**) est un objet indissociable, ayant une existence propre et un intérêt pour l'entreprise (par exemple client, fournisseur, article…).

Une **relation** (ou **association**) est une liaison sémantique entre plusieurs entités. Elle comporte autant de liaisons (appelées « pattes ») que d'entités en jeu dans la relation. Par exemple, la relation « commander » modélise une association entre les entités article et client (deux liaisons), tandis que la relation « livrer » modélise une association entre les entités article et fournisseur.

Un **attribut** (ou propriété) est la modélisation d'une information descriptive d'une entité ou d'une relation. Par exemple, le prix unitaire est un attribut de l'entité article, le nom est un attribut de l'entité client, la quantité commandée est un attribut de la relation commander et la date de livraison est un attribut de la relation livrer.

Ensuite, chaque entité doit être identifiable de manière unique. C'est pourquoi toutes les entités doivent posséder un attribut (ou un ensemble d'attributs) sans doublon possible (c'est-à-dire ne prenant pas deux fois la même valeur). Il s'agit de **l'identiant** que l'on souligne sur le schéma, par convention. Le numéro de client constitue un identiant classique pour l'entité client.

On remarquera qu'une entité possède au moins un attribut (son identiant) ; *a contrario* une association peut être dépourvue d'attribut. Elle possède cependant un identifiant implicite qui est l'ensemble des identifiants des entités en jeu dans la relation. Ce dernier, appelé identifiant de la relation, se matérialise dans les niveaux aval : modèle logique et physique.

La **cardinalité** d'une patte de relation entre une entité et une relation précise le minimum et le maximum de fois qu'une instance de l'entité peut être concernée par

l'association. Exemple : un client peut commander au moins un article et au plus *n* articles. On notera 1 : *n* la cardinalité de la patte côté client. De l'autre côté, un article peut ne faire l'objet d'aucune commande ou être commandé par plusieurs clients. On notera 0 : *n* la cardinalité côté article.

Dépendance fonctionnelle (ou CIF : « contrainte d'intégrité fonctionnelle ») : cette notion s'applique aux relations uniquement et exprime le fait qu'un individu de l'entité A ne correspond qu'à un et un seul individu de l'entité B. Par exemple, une commune n'appartient qu'à un et un seul département. Plus généralement, dans une relation binaire (deux entités en jeu), une cardinalité maximum égale à 1 induit une dépendance fonctionnelle.

On peut avoir aussi une **hiérarchie :** objets organisés selon une série de relations 1-*n* en cascade. Cette organisation de données est comparable à un arbre généalogique, où chaque membre n'a pas plus d'un père mais un nombre quelconque d'enfants.

La figure A.1 donne un exemple de MCD Merise.

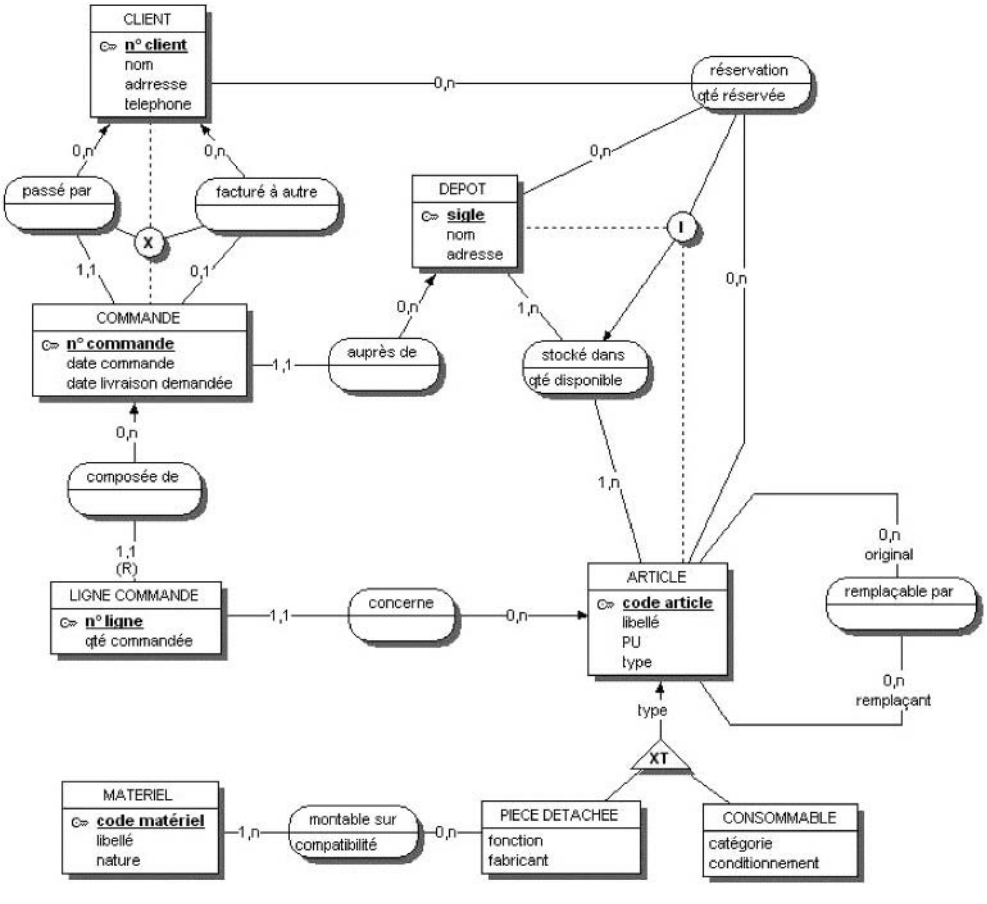

Figure A.1 – Exemple de MCD Merise

A.2.2 Modèle logique de données (MLD)

Lorsque des données ont la même structure (comme par exemple, les renseignements relatifs aux clients), on peut les organiser en **table** dans laquelle les colonnes décrivent les champs en commun et les lignes contiennent les valeurs de ces champs pour chaque enregistrement. Les lignes d'une table doivent être uniques, cela signie qu'une colonne (au moins) doit servir à les identier. Il s'agit de la **clé primaire** de la table.

On peut représenter les tables d'une base de données relationnelle par un schéma relationnel dans lequel les tables sont reliées par un connecteur. Toute entité devient donc une table dans laquelle les attributs deviennent les colonnes. L'identiant de l'entité constitue alors la clé primaire de la table. Une association binaire de type 1 : *n* disparaît au prot d'une **clé étrangère** dans la table qui référence la clé primaire de l'autre table.

A.2.3 Modèle physique de données (MPD)

Un modèle physique de données est le modèle de l'implémentation particulière du modèle logique de données dans un SGBD.

> Au contraire des modèles logique et conceptuel, le MPD dépend de la base de données et des détails de l'implémentation.

La traduction d'un MLD relationnel en un modèle physique est la création (par des requêtes SQL de type `CREATE TABLE` et `ADD CONSTRAINT`) d'une base de données hébergée par un SGBD relationnel particulier.

La figure A.2 donne un exemple complet de chaîne MCD/MLD/MPD.

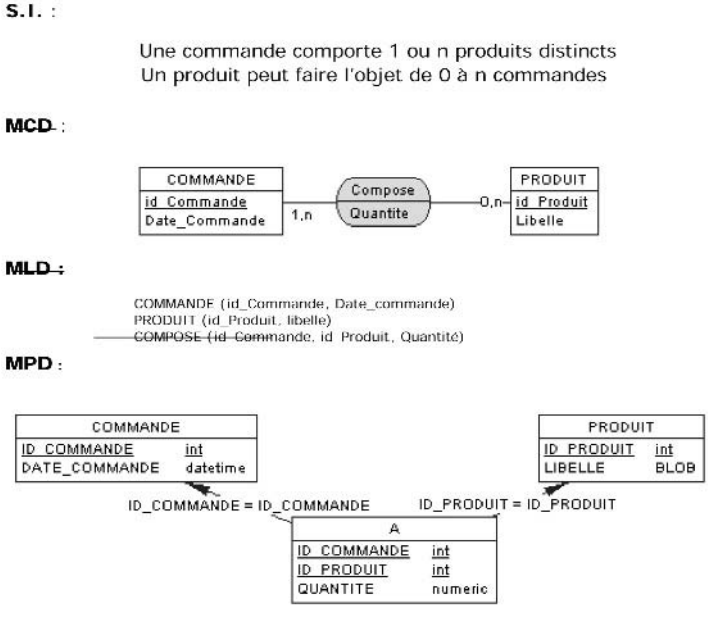

Figure A.2 — Exemple de chaîne MCD/MLD/MPD

A.3 UML

A.3.1 Notions UML

Une **classe** est le descripteur d'un ensemble d'objets qui ont une structure, un comportement et des relations similaires. Un objet représente alors une instance d'une classe ; il en possède les propriétés.

Un **attribut** est une caractéristique (ou propriété) d'un objet. L'ensemble des valeurs des attributs d'un objet constitue son état. À ne pas confondre avec son identité : deux objets distincts peuvent avoir le même état.

Une **association** est une relation sémantique entre deux classes ou plus. Elle se représente par un trait plein qui relie les classes en jeu. Une terminaison d'associations porte une **multiplicité** (domaine de valeurs de la cardinalité) qui s'exprime par un couple de valeurs n,p (n = valeur minimale, p = valeur maximale) ou une seule valeur si $n = p$.

On distingue trois types d'association :

- L'association simple.
- L'**agrégation** qui exprime une relation de type ensemble-élément : la suppression de l'ensemble n'induit pas la suppression de ses éléments. Exemple : une équipe est composée de joueurs, mais sa dissolution ne provoque pas la suppression des joueurs. Elle se représente par un losange vide sur la terminaison de l'association, côté ensemble.
- La **composition** qui exprime une relation de type contenant-contenu : la suppression du contenant entraîne celle du contenu. Exemple : si je supprime une facture, je supprime les lignes de facture qui la composent. Elle se représente par un losange plein sur la terminaison de l'association, côté contenant.

Une **généralisation** est un type de relation entre classes équivalant à l'héritage.

Les **classes d'association** sont des associations élevées au rang de classes. Elles portent donc des attributs.

Les **contraintes** expriment des restrictions sur des éléments de modélisation : classes, attributs, associations… Elles se formalisent par des notes reliées aux éléments *ad hoc* rédigées en langage naturel ou écrites en langage OCL (*Object Constraint Language*).

La figure A.3 donne un exemple de MCD UML.

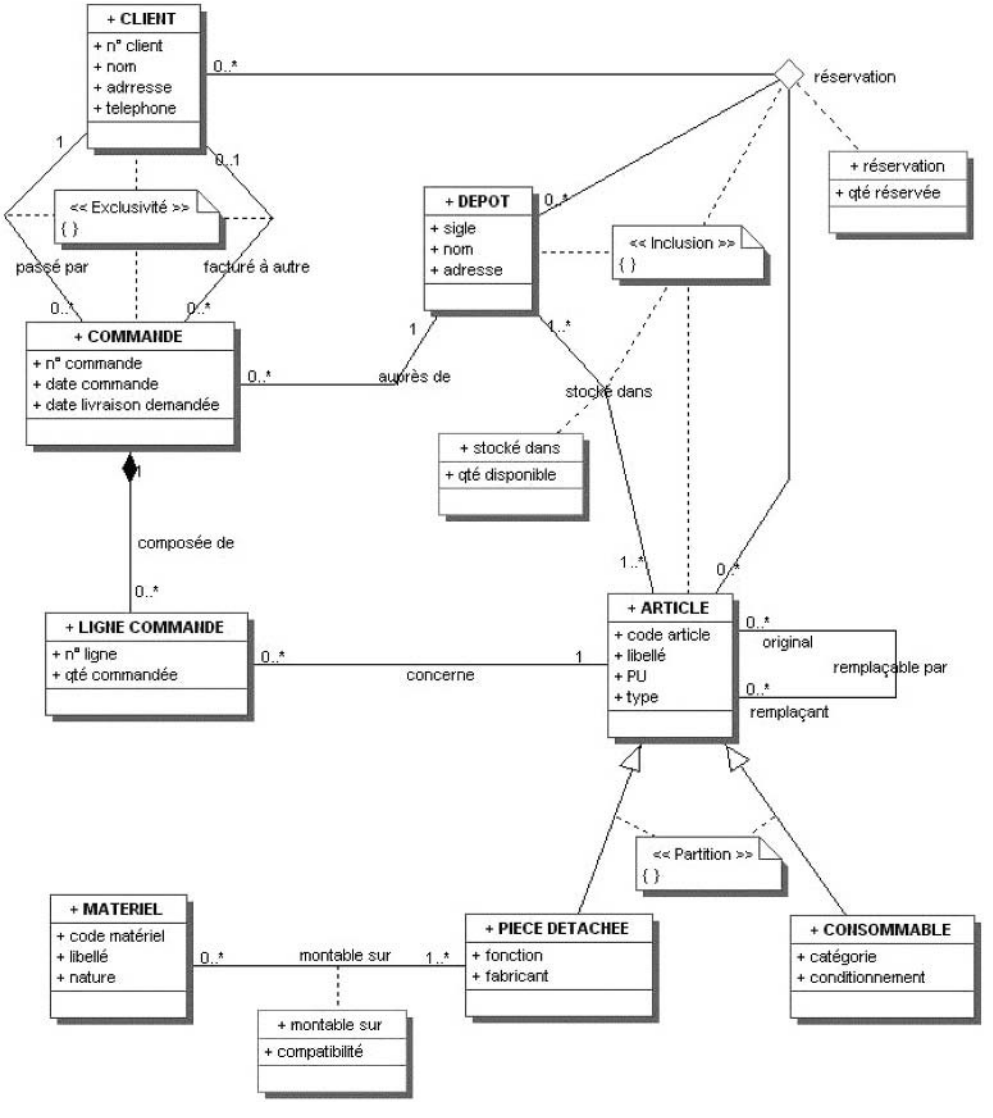

Figure A.3 — Exemple de MCD UML

A.3.2 Correspondance Merise UML

Toute **entité** est transformée en **classe**. Ses propriétés deviennent des attributs. Son identifiant devient un attribut identifiant ou **clé**.

Toute **relation** est transformée en **association**.

A.3.3 Modélisation XML

XML fournit plusieurs des caractéristiques que l'on retrouve dans les bases de données : le stockage (les documents XML), les schémas (XML Schemas…), des langages de requête (XQuery, XPath, XQL, XML-QL, QUILT…), des interfaces de programmation (SAX, DOM, JDOM) et ainsi de suite.

On peut ainsi décrire tout type de données avec les XML Schémas, et en particulier les tables d'un SGBD.On peut donc générer un schéma XML à partir d'un schéma relationnel et inversement.

Exemple de description d'un client en XML :

```
<Client>
    <Nom>ABC Industries</Nom>
      <Adresse>
      <Rue>123 Main St.</Rue>
      <Ville>Fooville</Ville>
      <Etat>CA</Etat>
      <Pays>USA</Pays>
      <CodePostal>95041</CodePostal>
    </Adresse>
</Client>
```

On peut lui associer une table avec les attributs nom et adresse.

A.4 LA MODÉLISATION DES FLUX DE DONNÉES

Pourquoi aborder spécifiquement ce sujet ? Aujourd'hui la modélisation des données en vue d'établir des modèles relationnels de SGBD est une activité qui décroît sensiblement dans les entreprises dans la mesure où les éditeurs de progiciels et de solutions complètes imposent leurs propres modèles de données. Les entreprises sont plus confrontées à la problématique de l'intégration desdits progiciels dans leur SI qu'à la construction de SGBD, **d'où l'importance croissante de la modélisation des flux de données dans les projets.**

A.4.1 Problématique

Comment les projets procèdent-ils en général ? Les interfaces sont traitées la plupart du temps au dernier moment et chaque flux fait l'objet d'une spécification propre *ex nihilo* validée par les producteurs et consommateur(s) du flux (dans le meilleur des cas, quand le producteur n'impose pas directement son format). Si on analyse *a posteriori* la structure et le format des données échangées pour l'ensemble des interfaces du projet, on constate une très grande disparité, entre les flux, des métadonnées utilisées :

- Un même élément source (objet, attribut, association) est implémenté sous différents éléments XML.

- Le type des données est variable d'une interface à l'autre : texte, numérique, nombre entier…
- Les données peuvent être à la fois obligatoires dans certains flux et conditionnelles dans d'autres.
- Les formats sont différents : il serait intéressant à cet égard de produire une liste « à la Prévert » pour tous les formats de date rencontrés, sans parler de la problématique heure légale/heure universelle (UTC).
- Idem pour les unités utilisées par les données de type montant ou mesure : on trouve en général tout le spectre entre les centièmes d'euro et les kilo-euros ou entre les watts et les méga-watts pour les mesures de puissance par exemple.

On peut identifier plusieurs raisons à cet état de fait dont les principales sont :

- L'absence de référentiel de métadonnées dans l'entreprise ou la direction.
- L'étanchéité des projets entre eux : aucune mutualisation ni partage d'informations notamment dans les échanges B2B.
- La multiplicité des intervenants (concepteurs, développeurs) : le fait que ce soient souvent des prestataires externes et de sociétés différentes n'arrange pas les choses.
- Les contraintes des progiciels, sans oublier les contraintes de coûts et de délais.

Bien sûr, la mise en œuvre de solutions urbanisées à base d'EAI et d'ETL permet de pallier ces disparités au prix de transformations appropriées mais elle ne résout pas le problème de la cohérence de l'ensemble. Elle ne fait qu'ajouter de la complexité et, de plus, génère ses propres métadonnées dont la plupart seraient inutiles si les projets partageaient un même référentiel de métadonnées.

Alors quelles peuvent en être les conséquences pour l'entreprise ? On citera principalement :

- Le temps de développement et de mise au point des échanges du fait d'une démarche *ex nihilo*.
- Les surcoûts des transformations inutiles de données en production.
- Les risques de sur ou sous-facturation des clients par mauvaise interprétation des données. Cela peut concerner les quantités de produit consommées ou les prix unitaires (erreurs sur les unités des montants et des quantités), le barème de facturation horo-saisonnier (erreurs sur les types de périodes de temps : heures creuses, heures pleines, pointe, hors pointe…) et donc une production comptable non fiable.
- Le report inévitable de ces imprécisions sur le SI décisionnel obligé d'harmoniser toutes les données publiées, de les rendre cohérentes. À cette étape, cela

coûte beaucoup plus cher, sans parler du risque de prise de mauvaises décisions basées sur des analyses erronées.

Pour anecdote, citons le fameux cas du projet de guerre des étoiles aux États-Unis, où la confusion entre mille nautique (1 852 m) et mile terrestre (1 609 m) par les uns et les autres (NASA, armée) a coûté des millions de dollars.

A.4.2 Les besoins

Face à cette situation et pour pallier ces risques, l'entreprise doit :

- Optimiser la charge et le délai de mise en œuvre des flux.
- Harmoniser leur modélisation.
- Synchroniser la documentation et les spécifications d'implémentation.
- Harmoniser l'utilisation des formats d'implémentation et notamment XML.

Elle doit viser, à cet égard, un objectif majeur : produire de façon automatique, c'est-à-dire véritablement industrielle, la documentation et les spécifications d'implémentation des flux (XSD, canonique EAI…), avec toute la qualité requise : cohérence, homogénéité, fiabilité et dans le respect des coûts et des délais.

A.4.3 Rappels sur l'urbanisme

En amont des projets, le plan d'urbanisme du SI formalise, à un horizon donné, une cible du SI composée notamment des grands blocs fonctionnels à outiller ainsi que les trajectoires pour atteindre ladite cible. Il identifie notamment les problématiques de couplage entre les sous-systèmes associés aux grands blocs fonctionnels.

Au niveau des projets, l'étude d'urbanisme, en conformité avec le plan d'urbanisme, et le dossier de pré-architecture technique précise le périmètre fonctionnel des sous-systèmes, les natures de couplages et les technologies des outils d'infrastructure à mettre en œuvre (EAI, ETL, services Web…) en fonction des solutions logicielles ou des progiciels choisis et des modes d'échanges (par lot, au fil de l'eau). Ainsi les projets ont une idée assez claire des différentes options qui s'offrent à eux pour chaque couplage à outiller, soit par exemple :

- Une solution *middleware* propriétaire pour les couplages internes à une solution progicielle, par exemple entre SAP CRM et SAP IS-U.
- L'EAI pour les messages unitaires liés à des couplages par événement, par exemple : la création d'un client dans le sous-système de gestion de la relation client déclenche un message distribué par l'EAI et destiné à mettre à jour les sous-systèmes de gestion des canaux automatisés (Web, téléphone).

- L'ETL pour les échanges de fichiers produits par le *batch* et liés à des couplages par les données, par exemple : la réplication dans un sous-système d'administration des ventes de la dernière grille de prix des produits/services produite par le sous-système de conception des offres.
- Les services Web pour les services entre sous-systèmes, par exemple la consultation d'un compte client sur le Web, une prise de RDV client dans le sous-système de gestion de la relation client *via* le sous-système qui gère le tableau de charge des techniciens dans le sous-système de gestion des interventions.
- L'échange de documents en solution « point à point » avec ou sans transformation.

A.4.4 Une méthodologie et des outils communs

Pour atteindre l'objectif d'industrialisation des flux, la bonne volonté ne suffit pas car le **problème est éminemment transverse** et les projets doivent impérativement adopter une méthodologie et des outils communs, exposés ci-après.

Tous les éléments cités le sont à titre d'exemple et chaque entreprise pourra évidemment les adapter, mais ils constituent la base même de la démarche.

Concepts

Préciser les concepts et fixer un vocabulaire commun est un préalable à toute méthode.

Peu importent les termes proposés ici, ce qui est important, c'est de bien distinguer les concepts suivants :

- *modèle de document* : structure pour échanger de l'information entre deux composants ;
- *document* : instance de modèle de document ;
- *flux* : ensemble des documents échangés relevant d'un même modèle ;
- *échange* : ensemble des flux d'un même processus.

Exemple : soit un processus de commande basé sur l'envoi d'une commande et la réception d'un accusé de réception de commande (ARC) :

- Les modèles de document sont les formulaires de commande et d'ARC.
- Les documents sont la commande et l'ARC.
- Les deux flux sont le flux des commandes et le flux des ARC.
- L'échange est constitué des deux flux commande et ARC.

À noter que le concept de document s'applique aussi bien à une commande unitaire émise lors du processus Commander qu'à un lot de commandes envoyé en *batch* fin de journée par exemple. Mais si le lot comporte un en-tête et une fin de lot, alors il s'agit d'un document différent de la commande avec son propre modèle.

Un modèle de document est en fait un diagramme de classes composé d'un ensemble **hiérarchique** de classes. Ces classes sont un ensemble d'objets qui partagent les mêmes attributs, relations, et sémantiques à l'exception des opérations et des méthodes (voir figure A.4).

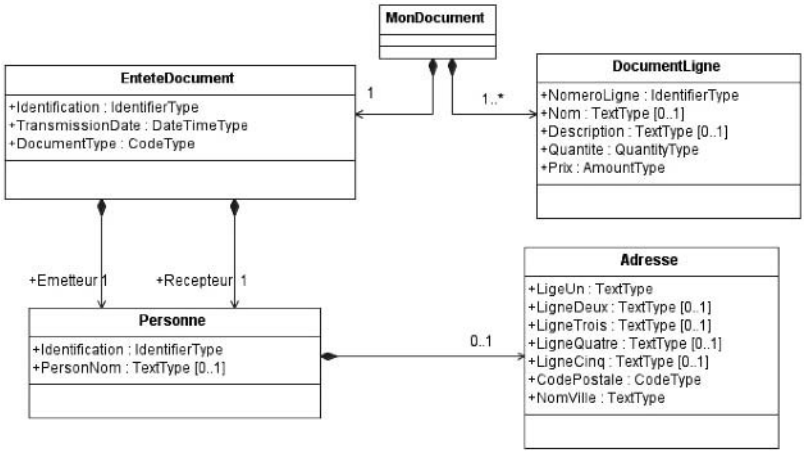

Figure A.4 — Exemple de modèle de document

Principes

Ils sont au nombre de trois :

- Se baser sur les normes et standards.
- Utiliser un seul langage de modélisation : UML.
- Utiliser une seule méthode de modélisation des documents quelle que soit leur implémentation (XSD…).

Méthode de modélisation

Elle s'appuiera sur :

- un catalogue unique de composants de base pour l'entreprise afin de garantir la mutualisation des objets métier décrits et l'homogénéité des modèles de document qui les mettent en œuvre ;
- une convention de nommage des classes et attributs ;
- des règles de construction UML des modèles de document ;
- des règles de génération XML ;
- une organisation *ad hoc* pour administrer l'ensemble.

Le « catalogue de composants » ou CCT (*Core Component Type*) propose un ensemble de types d'attributs prédéfinis qui se basent sur plusieurs types primitifs et

de composants agrégés (un composant agrégé est composé de plusieurs types primitifs). Exemple : une chronique est un ensemble de couples périodes/mesures.

Classiquement, les types primitifs sont au nombre de quinze : AmountType, MeasureType, QuantityType, DateTimeType, DateType, DurationType, PeriodType, NumericType, IntegerType, PositiveIntegerTyp e,BinaryObjectType, CodeType, IndicatorType, IdentifierType, TextType.

Tous ces types contiennent a minima l'attribut « content » accompagné éventuellement d'autres attributs, par exemple : AmountType contient un deuxième attribut 'currencyIdentifier' qui précise la devise dans laquelle le montant est exprimé.

les normes et standards à utiliser notamment sont :

- les normes ISO 4217 pour les unités de devises (AmountType), ISO 8601 pour les dates, dates heures et périodes ;
- les recommandations de l'UN/ECE[1]. Par exemple, la recommandation n° 20 pour les mesures (*cf*. le site http://www.unece.org/cefact/recommendations/rec_index.htm).

La « convention de nommage » des classes et attributs sera dérivée des principes décrits dans la norme ISO 11179 « Part 5 – *Naming and Identification Principles For DataElements* » avec les adaptations effectuées notamment par UN/CEFACT[2]. Cf. http://www.unece.org/cefact/xml/XML-Naming-and-Design-Rules-V2.0.pdf

La « construction d'un modèle de document » se base sur la méthodologie UMM[3] préconisée par l'UN/CEFACT, par exemple, sur un modèle d'information commun partagé par l'ensemble des projets d'un même métier. Ce modèle constitue un référentiel d'objets « souches » équivalents aux cellules souches utilisées en biologie. Les projets construisent ainsi leurs modèles de documents par dérivation de ce modèle commun et donnent toute garantie quant à la cohérence et l'homogénéité des structures d'échange et des informations échangées. La figure A.5 schématise cette méthodologie.

1. UN/ECE : *United Nations / Economic Commission for Europe*
2. UN/CEFACT : *United Nations Centre for Trade Facilitation and Electronic Business*
3. UNM : UN/CEFACT *Modelling Methodology*

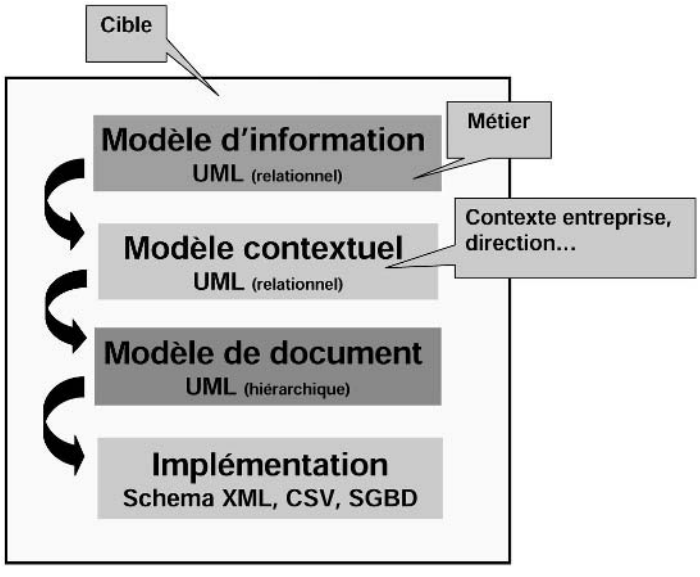

Figure A.5 — Méthodologie UMM

La construction se réfère au document de l'UN/CEFACT, *Requirements Specification Mapping*.

A.5 L'USAGE EN PRATIQUE DES MODÈLES DE DONNÉES

Les modèles de données sont des exemples de métadonnées. La connaissance de ces modèles est par exemple indispensable pour mettre en œuvre des échanges de données ou des migrations entre applications ne partageant pas les mêmes modèles.

Le MCD ou diagramme de classes est surtout mis en œuvre lors des phases d'urbanisme où l'on s'efforce de déterminer *a minima* les objets métier manipulés et leurs principaux attributs, ainsi que leur sémantique. Ces objets métier participent aux activités et processus métier. Ils sont souvent échangés entre activités (flux d'échanges).

Le modèle logique/physique est indispensable lors des phases de conception et d'implémentation d'applications, en particulier lors des échanges entre applications. En effet, il est parfois nécessaire d'effectuer des *mappings* (correspondances entre attributs de modèles différents) et des transcodifications (correspondances entre valeurs). Nous avons vu toutefois une approche modèle de flux visant à standardiser les modèles utilisés entre applications afin de limiter ces mappings et transcodages.

De même, la connaissance de ces modèles est indispensable lors des phases de migration entre applications.

De manière plus générale, la connaissance et la gestion de ces modèles sont un atout essentiel pour la gouvernance des données et permettent de maîtriser la chaîne de transformation de ces données. Dans le cas contraire, on se laisse imposer les modèles par les progiciels et on est conduit à réaliser un nombre sans cesse croissant de mappings et transcodifications entre applications (plus ou moins satisfaisants car ces correspondances ont nécessairement des limites). C'est pourquoi certaines entreprises ont défini un rôle à temps plein **d'architecte de données** (déjà évoqué dans le chapitre sur la gouvernance) chargé en particulier de maîtriser les modèles de données. Ces personnes connaissent la modélisation de données et leurs outils mais ils doivent aussi avoir une bonne connaissance des systèmes métier (ils doivent être spécialisés par domaines métier). Ils doivent par ailleurs avoir une capacité de dialogue/négociation avec les MOA/MOE et une très bonne capacité de synthèse et d'arbitrage.

B

SOA, services et données

Objectif

Il ne s'agit pas ici de décrire en détail les architectures orientées services (qui bénéficient par ailleurs d'une documentation abondante) mais de positionner la gestion des données de référence dans les architectures SOA, après de brefs rappels sur la SOA et les services Web. Le MDM peut être considéré comme une des briques de la SOA.

B.1 DÉFINITIONS

Le terme SOA est apparu en 1996 dans une note de recherche du Gartner : *« Service-oriented architecture (SOA) is a client/server software design approach in which an application consists of software services and software service consumers (also known as clients or service requesters). SOA differs from the more general client/server model in its definitive emphasis on loose coupling between software components, and in its use of separately standing interfaces »*.

Le SOA peut se décomposer ainsi :

- **S** comme « service » : fonctionnalités rendues par une entité pour une autre afin d'atteindre un résultat donné.
- **O** comme « orienté » : façon de concevoir l'architecture pour permettre à un ensemble de services d'interagir afin de satisfaire un besoin métier.

- **A** comme « architecture » : organisation d'un système à travers ses fonctionnalités et ses interactions vis-à-vis de son environnement.

SOA est donc avant tout **un paradigme d'architecture destiné à assurer l'interopérabilité, l'agilité et la réutilisabilité.**

Une architecture SOA ne va pas générer de valeur ajoutée en soi mais apporte un changement de perspective dans la construction des systèmes informatiques :

- Rapprochement vers les métiers :
  - C'est une approche du SI tirée par les processus qui remet donc la logique métier au cœur des fonctionnalités du SI.
  - Les services métier sont visibles donc mieux perçus par les responsables de processus : la démarche favorise l'implication des métiers dans la construction du SI.
- Simplification du SI vue de l'intérieur.
- Rationalisation et simplification du SI existant et du SI cible :
  - Le découplage entre applications, applications/données, données/données minimise les redondances et donc les risques d'incohérence.
  - La qualité intrinsèque de SI et son utilisation sont améliorées.
  - Les relations transverses entre SI sont optimisées (« interopérabilité » entre blocs applicatifs).
  - Les coûts sont réduits par réutilisation de composants (à l'intérieur d'un même bloc applicatif pour des composants métier ou de manière transverse pour des composants techniques).
- L'accès aux données est délocalisé, rendant transparente la synchronisation de bases hétérogènes et multiples.
- Les développements sont accélérés :
  - Les cycles de développements sont raccourcis : le développement de composants de services ré-utilisables focalise sur la construction de services nouveaux par combinaison d'éléments fonctionnels ou techniques existants (concept d'« application composite »).
  - Souplesse des modifications et meilleure intégration avec les autres systèmes.
  - Le coût de maintenance est plus faible.

La figure B.1 propose le modèle OASIS d'architecture SOA.

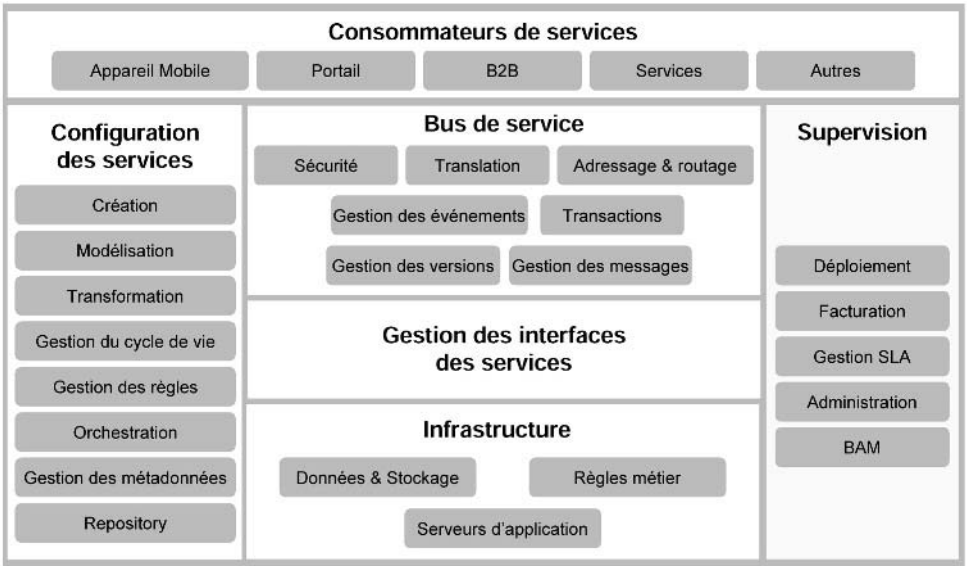

Figure B.1 – Modèle OASIS d'architecture SOA (Copyright Oasis)

B.2 OBJECTIFS ET ENJEUX DE LA SOA

Les systèmes d'information évoluent pour les raisons suivantes :

- Évolutions des organisations : il peut s'agir d'évolutions internes, comme par exemple la création de filiales ou de nouvelles implantations géographiques, la restructuration des organisations en place (centralisation des achats, des ressources humaines…) ou d'évolutions externes, comme les fusions ou acquisitions de sociétés.
- Évolutions réglementaires : ouverture à la concurrence des marchés publics (télécommunications, énergie, transport…), programme de décentralisation des services de l'État, lois sur la transparence… Autant d'obligations réglementaires qui obligent à repenser les systèmes d'information.
- Évolutions des métiers : les évolutions couvrent principalement la réaction face à la concurrence et à l'évolution de l'activité : mise en œuvre d'une gestion de la relation clients ou fournisseurs, distribution de services produits par des partenaires (services bancaires, services multimédias…) ou de « bouquets de services » personnalisés, ouverture à de nouveaux canaux de distribution.
- Évolutions technologiques : la technologie constitue la matière première des systèmes d'information. Que ce soit l'évolution des produits éditeurs, des modèles d'architecture (mainframe, client-serveur, *n-tiers*…), des paradigmes de programmation et leurs langages associés (Java, .NET…), les impacts sur le

système d'information sont nombreux et se traduisent par des efforts d'intégration entre socles technologiques hétérogènes.

- Maîtrise des coûts : c'est une évidence !

Quand un système informatique est amené à évoluer, on attend naturellement de lui qu'il facilite les évolutions : l'**agilité du système est donc une exigence majeure**. Dans cette optique, une **architecture modulaire de type SOA basée sur des services** est préférable à une architecture composée de blocs monolithiques. De plus, la standardisation autour des Web Services participe à **l'interopérabilité**.

> La valeur d'une architecture SOA se situe donc à la fois au niveau de la réutilisation de services mais aussi au niveau de l'interopérabilité.

Un des enjeux est donc la **réutilisation de services.** Cet enjeu prend tout son sens au niveau des services suivants :

- **Services d'infrastructure :** sans valeur métier mais que chaque application doit inévitablement mettre en œuvre pour ses besoins propres (annuaire, sécurité, échanges…).
- **Services métier de fine granularité :** si l'on prend l'exemple de l'invocation d'un service Web transverse de type « Validation d'adresse », la réutilisation d'un tel service évite la duplication de code entre applications ou entre modules de l'application.

En revanche, pour les services métier gros grain, l'enjeu porte principalement plus sur la **capacité de ces services à interopérer** avec les autres blocs du SI (mais bien entendu, l'interopérabilité est importante quelle que soit la granularité du service).

Ce double enjeu de la SOA, **réutilisation** et **interopérabilité,** peut être ainsi représenté figure B.2.

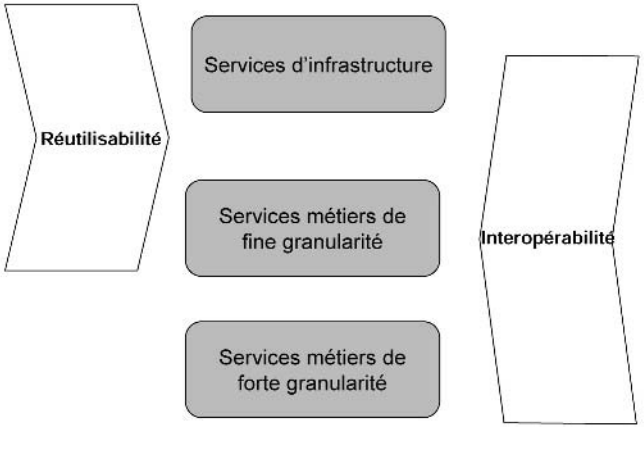

Figure B.2 — Principaux enjeux de la SOA

B.3 SERVICES

B.3.1 Définition d'un service

Le **service** est le composant clef de SOA. **C'est une boîte noire ayant une fonctionnalité bien définie**. Autonome, il ne dépend d'aucun contexte ou de service externe. Le service échange des messages avec l'extérieur (analogie avec la lettre et son enveloppe). La communication entre les services peut consister en un échange de messages ou en une conversation (plusieurs échanges).

Derrière cette définition, se cache **la notion de contrat de service**. Un service se doit d'assurer l'action qui lui est demandée et qu'il effectue. Il assure par ce biais un contrat d'interface et une qualité de service, ces deux aspects formant le contrat de service. Par exemple, un service de calcul doit assurer une disponibilité minimale et une fraîcheur d'information définie par avance. Un service d'authentification unifiée se doit d'assurer une disponibilité à toute épreuve.

Le contrat de service porte sur le traitement demandé et son intégrité. Quand nous évoquons le traitement associé à un service, il peut s'agir de plusieurs opérations. Par exemple, un service de gestion de client peut être constitué des opérations `créer client`, `supprimer client`, `modifier client`. Le contrat de service est donc avant tout **un contrat d'interface, qui doit être stable dans le temps** (voir la figure B.3).

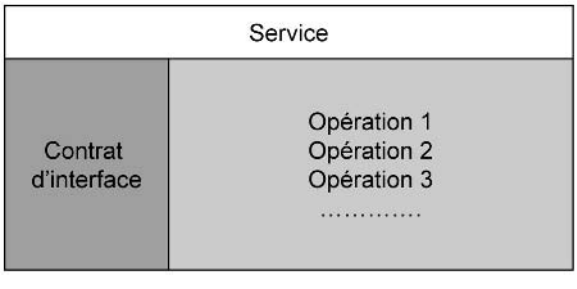

Figure B.3 — Service et contrat de service

L'interface d'un service est décrite à travers des paramètres d'appels, des paramètres de retour et des conditions d'erreur. **Des données peuvent être transférées à travers ces paramètres. Le service peut lui-même accéder à des données pour effectuer les traitements demandés** (voir la figure B.4).

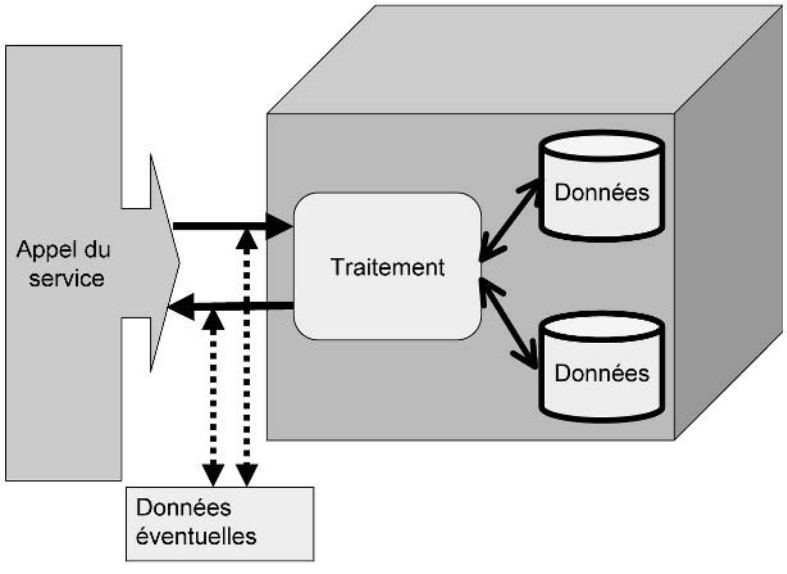

Figure B.4 — Logique de service et données

Les caractéristiques importantes d'un service sont les suivantes :

- Couplage lâche (*loose coupling*). Le consommateur et le producteur sont gérés et contrôlés par des entités différentes. Les changements apportés à l'un des participants n'ont pas d'influence sur le deuxième. On peut construire l'un sans connaître l'autre et indépendamment des technologies. Ce couplage lâche est renforcé par la mise en œuvre d'un bus ESB (*Entreprise Service Bus*).
- Interface : c'est le point d'entrée et de sortie du service. Utilisé par les consommateurs, il est adressable sur le réseau. Il définit les limites de communication entre les services. Il offre un mécanisme d'abstraction du service envers l'extérieur.
- Synchronisme : la communication peut être synchrone (attente de la réponse du service) ou asynchrone.
- Composition et encapsulation : les opérations proposées par un service métier encapsulent plusieurs fonctions et opèrent sur un périmètre de données large, contrairement à la notion de composant technique type EJB, *portlet*…

Des règles ont été définies par l'organisme OASIS :

- Un service consiste en un ensemble de fonctionnalités fournies par une entité dans le but d'être utilisées par d'autres entités.
- Les services sont conceptuellement autonomes (autosuffisants) et indépendants des technologies d'accès aux ressources.
- Aucune distinction en terme d'architecture ne doit être faite entre les services consommés dans le cadre d'un processus et les autres.
- Tout service logique aura une description canonique.

- La description d'un service est composée de trois parties logiques :
  – un modèle de données ;
  – des règles et obligations exigées pour les consommateurs et les fournisseurs du service ;
  – un contrat qui gouverne l'utilisation du service.
- Une règle de sécurité peut nécessiter l'utilisation d'un système de sécurité.
- Une politique de sécurité inexistante est considérée comme une règle de sécurité.
- Les consommateurs d'un service doivent disposer de sa description et la qualité de service proposée afin de pourvoir interagir au mieux avec ce dernier.
- La découverte d'un service est différente de l'autorisation d'exécution.

Un service a pour vocation initiale l'exposition de traitements. Dans ce sens, il apporte une décorrélation entre fournisseur de service et consommateur de service. Cela a pour objectif d'apporter un niveau d'abstraction supplémentaire aux traitements effectués. De cette manière, un service va pouvoir être utilisé par plusieurs types de consommateurs. Un service peut par exemple être utilisé pour un traitement *batch* massif et pour un traitement « à la demande » par un utilisateur. Par ailleurs, un service peut également être invoqué par un autre service.

Il importe par ailleurs de bien distinguer fonction, composant, service et processus :

- Une fonction correspond à un sous-programme ou une classe d'objet.
- Un composant correspond à une unité de traitement exécutable (EJB pour *Enterprise Java Bean*), *servlet*…
- Un processus correspond à un assemblage de services orchestrés.
- Les services gèrent messages, données et composants**. Les données privées sont totalement encapsulées par le service**. Les messages sont le seul moyen d'échanges entre services.

B.3.2 Accès aux services

L'accès à un service peut être direct ou via un bus de service (appelé **ESB** : ***Enterprise Service Bus***). Ce dernier permet l'accès à un service via un connecteur, l'ESB assurant la couche de transport (voir la figure B.5).

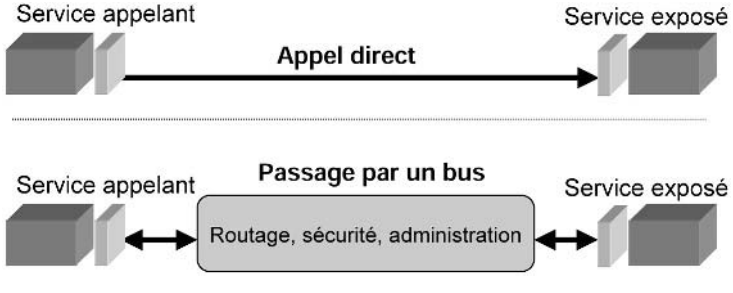

Figure B.5 — Accès aux services

L'intérêt d'un bus de service est de permettre le routage, l'administration et la sécurisation de l'appel aux services, mais aussi d'assurer un couplage lâche entre consommateurs et producteurs de services (un service peut être invoqué via une interface proxy, indépendante de l'interface réelle).

B.3.3 Services et services Web

Les services Web (ou *Web services*) sont un type d'implémentation standard des services. Ils reposent principalement sur l'utilisation d'interfaces d'invocation et de vocabulaire de description de données standardisés, qui doivent donc être communs à l'ensemble des agents (fournisseurs de services et utilisateurs de services).

En termes d'interopérabilité, les architectures SOA basées sur les services Web reposent sur les standards décrits grâce au WS-I (*Web Service Interoperability*). Parmi les différentes couches de normes et protocoles qui permettent de bâtir de telles architectures, on relève :

- La gestion d'un annuaire de services (quels sont les services mis à disposition et par qui) avec **UDDI** (*Universal Description Discovery and Integration*).
- La description des interfaces des services (quelles sont les données nécessaires à l'exécution du service, que fournit-il en retour…) avec **WSDL** (*Web Services Description Language*).
- L'invocation (ou l'appel) du service (la requête transmise au service) avec **SOAP** (*Simple Object Access Protocol*) ou **REST** (*Representational State Transfer*), qui spécifie en fait l'utilisation de HTTP et XML.
- Le format des données échangées avec **XML** (*eXtensible Markup Language*).
- Le transport des données avec les protocoles Internet HTTP et TCP/IP. Une architecture SOA peut être également complétée par :
  - La gestion de la sécurité avec XML Signature, XML Encryption, SAML (*Security Assertion Markup Language*) ou encore XKMS (*XML Key Management Specification* qui gère les infrastructures à clé publique ou PKI).
  - L'orchestration (on parle également de chorégraphie) des services pour constituer des processus métier avec BPEL4WS (*Business Process Execution Language For Web Services*) devenu WS-BPEL.
  - La gestion transactionnelle…

La figure B.6 résume ces principaux standards.

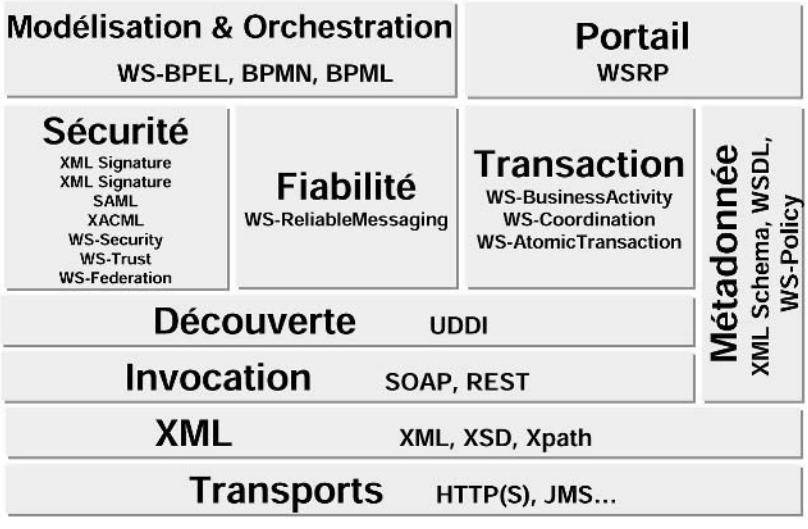

Figure B.6 — Standards des services Web

B.4 SOA EN PRATIQUE

Quand on évoque quelques dizaines de services, souvent des **services Web** développés pour un système, on a affaire à du **SOA technique**. Dans ces cas, le système, les applications, ne changent pas. Le système (ou des portions du système) est juste enrobé d'une couche de services **qui en facilite l'accès**.

Ce SOA technique est une étape qui ne permet pas toujours d'améliorer la qualité globale des systèmes, de réduire la redondance interne et d'augmenter la réutilisation. Atteindre ces objectifs exige un effort de conception des services qui passent par une étude d'urbanisme adéquate.

Mais en pratique, cette étape cible de la SOA est encore assez peu mise en œuvre dans les systèmes d'informations d'entreprise. Par contre, améliorer les mécanismes d'échange via des services Web tend à se répandre. C'est un moyen d'échange « au fil de l'eau » qui permet de mieux synchroniser les applications par rapport à des échanges classiques de types transferts de fichiers *batch*.

B.5 SOA ET INTÉGRATIONS DE DONNÉES

Sous sa forme la plus simple, l'architecture orientée service (SOA) exige que les fonctions d'une entreprise (celles qui étaient contenues dans des applications distinctes) soient prises en compte dans un ensemble de services métier. Ces services

communiquent avec des clients et d'autres services internes et externes en recevant des messages de requête de services (« Je veux le service X ») et en envoyant des messages de réponse (« Voici la réponse Y »).

Toutefois, lorsque les architectes et les développeurs commencent à réfléchir à la conception et à l'implémentation d'une SOA, des questions surgissent : **quelles données les services doivent-ils partager ? Comment accède-t-on à ces données ? Comment sont-elles gérées et représentées ?**

On peut distinguer deux types de données dans une architecture SOA :

- Les données de messages (paramètres des messages).
  Les données qui transitent entre des services sont appelées données de messages. Ces données sont connues à travers l'interface du message décrite en WSDL.
- Les données métier ou techniques.
  C'est le type de données auquel pensent la plupart des personnes lorsqu'elles parlent d'applications (informations client, stock de produits et informations bancaires, par exemple). Les données métier sont encapsulées par des services. Elles sont en général stockées dans un SGBD.

La gestion des données de référence accélère et sécurise les processus de consultation et de mise à jour des données, en particulier celles dupliquées dans plusieurs systèmes.

> Dans cette approche, plutôt que de propager les flux de mise à jour en mode « point à point » ou via un broker d'intégration, on préfère bâtir un référentiel pour le stockage des données, puis propager les modifications à partir de ce point unique.

Ainsi, on assure l'exactitude de la valorisation de chaque donnée de référence au sein de chaque système qui la consomme, sous condition de disposer aussi d'une solution d'intégration des données qui garantisse le bon acheminement des modifications. À ce titre, l'architecture orientée services renforce l'efficacité de la solution en connectant de manière standard le référentiel aux systèmes de production et décisionnel.

Dans cette démarche**, le MDM devient une brique d'une architecture SOA**. En effet, disposer d'une architecture orientée services exige des données disponibles et à jour, pour que les services et les processus qui y font appel donnent les résultats escomptés. **L'accès aux données peut alors s'effectuer « au fil de l'eau » (et non en *batch* comme dans beaucoup d'architectures anciennes) *via* des services qui sont implémentés sur le progiciel MDM.**

La figure B.7 schématise la mise en œuvre d'un référentiel de données (type MDM) dans une architecture SOA.

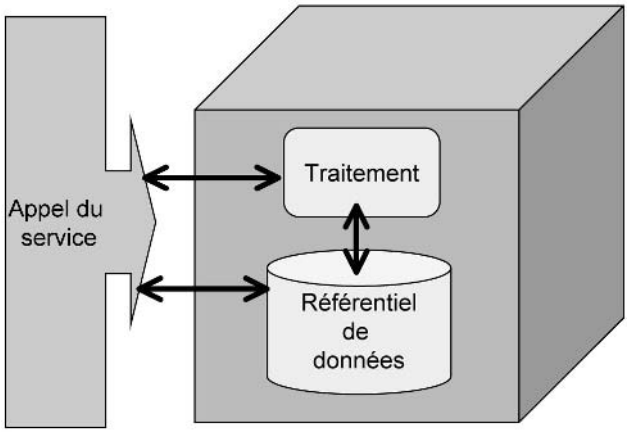

Figure B.7 — Référentiel de données dans une architecture SOA

Glossaire

Acteur – Personne qui agit sur un système. Un acteur possède un ensemble de droits d'action sur le système (dénommé profil) selon les rôles qu'il exerce.

Activité – Ensemble **d'actes coordonnés et de travaux réalisés par un ou plusieurs êtres humains au sein de l'entreprise**. Le découpage des activités dans l'entreprise met en évidence des métiers différents (Vendre, Produire…). Les activités sont décrites par décompositions successives.

Administration des données – L'administration de données est la fonction de l'entreprise qui définit et maintient la sémantique des données, les formalise, les communique et en contrôle l'usage par les autres fonctions.
Par extension, désigne **toute activité liée à la gestion des données**, pouvant aller jusqu'à l'administration de base de données.

Annuaire – Un annuaire (*directory*) est un système de stockage de données, dérivé des bases de données hiérarchisées, permettant en particulier de conserver les données pérennes, c'est-à-dire les données **n'étant que peu mises à jour (historiquement, sur une base annuelle, d'où le nom), comme les coordonnées des personnes, des partenaires, des clients et des fournisseurs d'une entreprise**. C'est pourquoi, grâce à des optimisations, un annuaire est beaucoup plus rapide en consultation qu'en mise à jour.

Annuaire de services – C'est une application qui permet de **stocker les descriptions des services et les met à la disposition des consommateurs**. On va généralement utiliser le standard WSDL pour décrire les services.

Application – Produit informatique développé pour répondre à un besoin spécifique ou développé par un éditeur de logiciels.

Applications analytiques – Applications fournissant à **l'entreprise des indicateurs et des données lui permettant de suivre les tendances de ses activités.** Entrent dans cette catégorie tous les programmes et logiciels qui analysent les données relatives aux activités opérationnelles et aux clients de l'entreprise, et les présentent sous une forme permettant des prises de décision efficaces et rapides.

Architecte des données – L'architecte est un rôle dans la filière SI dont les missions sont les suivantes :
– Capitaliser les modèles de données (y compris leur sémantique) et assurer leur cohérence.
– Décrire et capitaliser les flux de données et les *mappings*.
– S'assurer que les données sont pertinentes, correctement utilisées, efficacement partagées et du niveau de qualité souhaité.
– Aider le métier à appréhender la gestion des données et les technologies associées.
– Résoudre et anticiper les conflits autour des données et détecter les opportunités d'amélioration (exemple : mise en cohérence, évolutions des référentiels, amélioration de la qualité…).
– Assurer la conformité aux standards du métier et du SI, aux impératifs réglementaires sur les données et aux règles de sécurité.
En termes de profil, il a une compréhension des processus métier qui manipulent les données et connaît les applications correspondantes. Il maîtrise les outils de modélisation, les technologies de gestion des données et de qualité des données.

Architecture de centralisation MDM – Une architecture de centralisation MDM repose sur le support direct des processus référentiels (création/modification/suppression) par la solution de gestion des données maître. **Le point d'acquisition et le point de vérité sont ainsi fusionnés au sein de la solution**.

Architecture de consolidation MDM – Une architecture de consolidation MDM s'entend quand plusieurs sources de données alimentent le référentiel. Ces sources de données sont indépendantes de la solution et des applications qui s'y alimentent. **Les points d'acquisition de la donnée sont, dans cette architecture, les flux d'alimentation de la solution**.

Architecture de coopération MDM – Architecture qui utilise les applications existantes comme points d'acquisition de la donnée. Ces applications sont partie intégrante de la solution car :
– Les processus référentiels (création/modification) sont partagés entre ces applications et le référentiel.
– Elles sont dépendantes du référentiel afin de pouvoir utiliser les données dont elles sont sources (préservation du référentiel comme source de vérité et protection contre les risques de désynchronisation des processus entre applications source et consommatrices).
Ainsi **chaque donnée saisie dans l'application source descend vers le référentiel puis le référentiel renvoie une validation à l'application source.**

Architecture de répertoire virtuel – Architecture correspondant à une architecture de Consolidation. Cependant elle **utilise une technologie EII** afin de mettre à disposition l'information aux applications consommatrices plutôt qu'un référentiel.

Association (ou **relation**, ou lien) – Association qui définit **un type de lien, un type de relation entre deux ou plusieurs entités.** Par exemple, on établit une association entre un client et un contrat, qui sont deux entités séparées.

Attribut (ou propriété) – D'un point de vue conceptuel, un **attribut** (ou une **propriété**) est une caractéristique associée à une entité ou objet (ou encore classe). Par exemple, un client a pour attributs : nom du client, adresse, numéro… Un attribut prend ses valeurs dans un domaine.
Du point de vue logique et physique, **un attribut est un identificateur (un nom) décrivant une information stockée dans une base**. Par exemple, le numéro de sécurité sociale et le nom d'une personne sont des attributs d'une personne.
Un attribut dont la valeur particulière (occurrence de l'attribut) permet d'identifier une occurrence de l'entité est **l'identifiant** de cette entité (appelé aussi **clé**).

Back office – En architecture informatique, le ***back-office*** est la partie gestion non visible de l'utilisateur, opposée au *front office* qui est la partie dédiée à l'interaction avec l'utilisateur. Il s'agit des bases de données, des moniteurs transactionnels, des processus de traitement internes, ainsi que de toute application qui ne requiert pas l'intervention de l'utilisateur final.

BAM (*Business Activity Monitoring*) – Concepts et outils permettant **de piloter en temps réel les processus métier de l'entreprise**. Le BAM fournit un tableau de bord, des indicateurs de mesure de la performance métier, des outils de *monitoring*, de *reporting* et permet le contrôle du rendu fonctionnel des processus métier.

Banque de données – Ensemble de données relatif à un domaine défini des connaissances et organisé pour être soumis aux consultations d'utilisateurs.

Base de connaissances (*knowledge base*) – Ensemble d'informations, en particulier des règles et des faits, qui constituent **le domaine de compétence d'un système.**

Base de données – Ensemble de données organisé en vue de son utilisation par des programmes correspondant à des applications distinctes et de manière à faciliter l'évolution indépendante des données et des programmes.

BI (*Business Intelligence*, ou décisionnel) – Techniques d'organisation et outils de travail pour la gestion des informations et des documents de **pilotage de l'entreprise**. La *Business Intelligence* comprend principalement la structuration des indicateurs de pilotage au sein d'un entrepôt de données (*data warehouse*), l'exploration de ces données (*data mining*) et la restitution de ces analyses au travers de tableaux de bord.

BPM (*Business Process Management*) – Un logiciel de BPM permet **d'ordonnancer des activités ou des processus**. BPM est un terme générique désignant à la fois la modélisation et la traduction, à l'aide d'un « moteur », des processus modélisés dans la réalité de l'entreprise. À cela s'ajoutent des outils – indicateurs, tableaux de bord… – de suivi de l'exécution des processus et d'évaluation de leur performance (voir BPMS).

BPMS (*Business Process Management System*) – Ensemble logiciel destiné à formaliser les procédures qui font l'activité d'une entreprise dans le but de les automatiser. Cet ensemble comprend généralement :
1. Un outil de modélisation qui sert à formaliser la description des fonctions exercées dans l'entreprise en processus, en applications informatiques. Il permet de définir également les données échangées, les interfaces avec les autres modules.
2. Des outils de développement pour formaliser la logique qui régit les processus de l'entreprise.
3. Un moteur d'exécution qui supervise le déroulement des processus ainsi que les échanges de paramètres.
4. Un moteur de règles qui évalue l'état de tous les objets impliqués dans le déroulement des processus et détermine si les conditions sont remplies pour en lancer, poursuivre ou arrêter l'exécution.
5. Un référentiel qui mémorise tous les objets manipulés, en particulier les définitions des processus, les règles qui doivent déclencher leur exécution, les contraintes d'intégrité, de sécurité ainsi que les mesures de référence relatives au métier de l'entreprise.
6. Des outils d'administration qui permettent de régler les paramètres de l'ensemble du système et d'obtenir des indicateurs de performance et des statistiques à partir des données collectées lors de l'exécution des processus.

CDI (*Customer Data Integration*) – Combinaison des technologies, processus et services nécessaires pour créer et maintenir une **vue exacte, complète et à jour du client** quels que soient les canaux de distribution de la donnée, et l'organisation utilisatrice, et cela dans un contexte où plusieurs sources de données client coexistent.

Client léger – Variante de la composante client d'un modèle client-serveur, dans laquelle **le volume du traitement effectué sur l'ordinateur client est minimal** en comparaison du volume traité sur le serveur. Dans une architecture client léger-serveur lourd, l'ordinateur client pourrait à la rigueur n'accueillir qu'un logiciel d'interface graphique lui permettant d'afficher uniquement le résultat des traitements effectués sur le serveur.

Client lourd – Dans une architecture client-serveur de première génération, poste client constitué d'un micro-ordinateur traditionnel, doté d'un système d'exploitation et **d'applications installées sur le disque dur local.** Un client lourd peut être un PC traditionnel, une station de travail haut de gamme ou un ordinateur portable. Les clients lourds peuvent fonctionner connectés au réseau ou de façon autonome. Ils présentent des avantages de performance, de souplesse et de mobilité, mais nécessitent une maintenance difficile et onéreuse.

Client riche – Est relatif à l'architecture client enrichi dans laquelle le **poste client assume une bonne partie des traitements,** alors qu'un serveur Web sert de relais entre le poste client et une base de données située sur un autre serveur.

Cohérence – La cohérence des données à l'intérieur d'un SI consiste à disposer des **mêmes valeurs d'attributs concernés pour l'ensemble des applications du SI.**

Complétude – Terme relatif à la qualité des données : **toutes les informations requises sont-elles renseignées ?**

Connecteur – Désigne un élément informatique qui permet une **prise sur un logiciel** afin de lire ou d'écrire des données, de réaliser des traitements. On évoque en particulier les connecteurs pour les bus d'échange, mais aussi pour le BAM, le BPM…

Connectivité – Capacité d'un programme ou d'une machine à se **connecter à d'autres programmes/machines.** Par exemple, on dira d'un programme capable d'importer les données de nombreux autres programmes qu'il a une bonne connectivité.

Contrat d'échange – Engagement sur la **disponibilité et la qualité des données**.

Contrat d'interface – Contrat spécifiant les **entrées, sorties, conditions d'erreur**.

Coordinateur des données (ou intendant) – Il s'agit normalement d'un intervenant métier, mais ce rôle peut être délégué à un intervenant SI.
Le coordinateur (*data steward,* terme que l'on peut aussi traduire par intendant) gère la sécurité et la qualité définies par le propriétaire. Il définit les droits d'accès et de diffusion, spécifie les critères de qualité requis, prépare les indicateurs de suivi, puis gère la mise en œuvre et le suivi opérationnel. Il peut être éventuellement chargé de la validation des données créées et de leur diffusion. À tout le moins, il peut aider les utilisateurs à créer, modifier et supprimer les données.

CRM (*Customer Relationship Management*, ou GRC : gestion de la relation client) – Ensemble des technologies, de l'organisation et des méthodes plaçant la **relation client et sa satisfaction au cœur de l'entreprise**, et se traduisant par la mise en place d'outils logiciels spécifiques. Permet la gestion dynamique de la relation client tous médias (téléphone, Internet, mail, fax, lettre…) et tous les services de l'entreprise (ventes, marketing, après-vente). On dissocie trois axes :
– CRM analytique qui concerne la collecte et l'analyse en profondeur des données clients.
– CRM opérationnel qui planifie l'action commerciale utilisant un canal (téléphone, mail…).
– CRM collaboratif qui optimise et concerte les actions sur les différents canaux d'interaction avec la clientèle.

Cycle de vie métier de la donnée – Le cycle de vie métier de la donnée correspond à **la séquence des états métier possibles de cette donnée**. En effet, la donnée ne répond pas aux mêmes règles (attributs à saisir, droits…) à chacun de ses états. Le passage d'un état à un autre est déclenché par un « événement métier ». Exemples d'états pour un objet client : prospect, demandeur, client actif, ancien client.

Cycle de vie technique – On peut décrire un cycle de vie technique de la donnée par les **étapes par techniques lesquelles passe la donnée (création, modification…).**

***Data cleansing* (nettoyage des données) –** Opération par laquelle on s'assure que les données d'un fichier sont cohérentes, non dupliquées, conformes à une référence, complètes…

***Datamart* (magasin de données) –** Base de données conçue pour aider les opérationnels dans leurs décisions stratégiques. Tandis que les *data warehouses* associent des bases de données diverses, les *datamarts*, généralement de plus petite taille **se concentrent sur un sujet ou sur les activités d'un métier spécifique**.

***Data profiling* (profilage des données) –** Fournit des informations sur le **contenu, la qualité et la structure des données** de tous types de systèmes opérationnels de façon à permettre une intégration de données efficace (utile aussi pour la migration des données).

Data set – Représente une **collection de données** regroupées de manière logique.

***Data warehouse* (entrepôt de données) –** Base de données conçue pour servir de support aux applications décisionnelles. Les *data warehouses* contiennent une grande diversité de données permettant d'avoir une image cohérente de l'activité à un moment donné. Le data warehousing consiste à **agréger les données de plusieurs bases de l'entreprise dans une nouvelle base partagée.**

Dictionnaire de données – Le dictionnaire des données (**type de métadonnées associé à un SGBD**) se présente sous la forme d'un tableau. Dans ce tableau, chaque donnée est représentée par :
– son nom informatique, c'est-à-dire un mnémonique ou un nom en clair ;
– une description ;
– son type numérique, alphabétique, logique…
– sa dimension en nombre de caractères ;
– éventuellement les règles de gestion associées relatives à son élaboration, la cohérence avec d'autres données…
– le domaine de valeurs.

Domaine de valeurs – Le **domaine** de valeurs d'une donnée ou d'un attribut **est l'ensemble, fini ou infini, de ses valeurs possibles.** Cet ensemble peut se présenter sous la forme d'une liste de valeurs, d'un intervalle de valeurs, d'une contrainte particulière (par exemple le caractère positif des valeurs)… Par exemple, le domaine de valeurs de la propriété « Mois » est une liste de douze valeurs : 01, 02... 12.

Donnée (data) – Description élémentaire de nature numérique ou alphanumérique, **représentée sous forme codée** en vue d'y être enregistrée, traitée, conservée et communiquée et qui est compréhensible par la seule machine.
Ne pas confondre modèle de la donnée (attributs de l'objet) et valeur (valeurs de l'instance, et donc des attributs). Lorsqu'on ne précise pas, donnée signifie valeur de la donnée.

Données structurées et non structurées – Les données peuvent être stockées dans un de ces deux types – structuré et non structuré. Les données **structurées se rapportent aux informations qui sont stockées dans un format répété et structuré**. Les données structurées se rapportent à des dossiers, à des tables, à des bases de don-

nées, à des entrepôts de données. Des programmes informatiques facilitent le stockage et l'accès à ce type de données.
Les données **non structurées sont généralement présentées sous un format plus facile à comprendre pour un cerveau humain (par opposition à une machine).** Les données non structurées incluent des documents, des images, des graphiques, des vidéos, des fichiers audio…

Donnée de référence ou donnée maître ou *master data* – Une donnée de référence est une donnée à laquelle il est fait référence dans plusieurs applications. C'est une référence pour une activité de l'entreprise.
La donnée de référence doit être de qualité et respecter des normes communes qui s'imposent à l'ensemble des entités utilisatrices.

Données « constitutives » – Ce sont les données de référence qui vont servir **d'attributs aux données maître**.
Elles possèdent elles-mêmes plusieurs attributs. Exemples : adresses, communes.

Données « maître » – Ce sont en général **les objets métier principaux** (« cœur de métier ») d'un domaine fonctionnel. Sont au cœur du SI du système fonctionnel et font l'objet des principales applications. Exemples : client, article, fournisseur, point de mesure…

Données « paramètre » – Ce sont **des tables de valeurs ou des nomenclatures**. Données qui donc entrent en jeu comme source dans une contrainte de référence. Exemples : codes postaux, codes devises, taux des taxes des communes.

Données décisionnelles – Ce sont des **données consolidées permettant des analyses** statistiques et l'édition de rapports afin de piloter l'activité d'entreprise.

Données transactionnelles (ou opérationnelles ou de production) – Les données transactionnelles sont les **données résultantes du déroulement d'un processus métier.** Elles représentent l'activité de l'entreprise.

DQM (*Data Quality Management*) – Le DQM ou gestion de la qualité des données est une suite de solutions qui permet de répondre à **l'amélioration du contenu des données** en assurant les fonctions de :
– détection des erreurs de typage, de format, de contenu des données et les doublons ;
– mise en œuvre de processus de gestion de la qualité : correction, standardisation, complétion et consolidation.

Duplication – Terme relatif à la qualité des données. **Existe-t-il de multiples et inutiles représentations des mêmes objets** de données dans l'ensemble des données ?

EAI (*Enterprise Application Integration*) – Bus de communication permettant des échanges de messages entre applications, permettant d'éviter la multiplication des échanges « point à point ».

EAM (*Enterprise Asset Management*) – Désigne un **progiciel destiné à gérer les actifs** (locaux, produits, équipements…) d'une entreprise.

ECM (*Enterprise Content Management*) – Ensemble organisé de ressources matérielles, logicielles, humaines, informationnelles permettant de collecter, traiter, stocker et distribuer **tous les types d'informations (structurés et non structurés) nécessaires à une entreprise**.

EII (*Enterprise Information Integration*) – Catégorie de logiciels qui permet **l'intégration de données éparpillées dans une base de données virtuelle**. Cette infrastructure permet ainsi à une application métier d'accéder à un ensemble de bases par le biais d'une vue logique unique.

EIM (*Entreprise Information Management*) – Consiste en un framework permettant la réconciliation sémantique des données. EIM utilise en ce sens plusieurs outils, technologies et techniques, garantissant aussi l'interopérabilité des systèmes.
L'EIM recouvre des champs tels que :
– le MDM (incluant la qualité de la donnée) ;
– les *metadata repositories* et les *metadata registries* ;
– les solutions d'échanges inter-applicatives (EAI, ETL, BPM…) ;
– les solutions d'indexation et de recherche d'information ;
– les solutions d'anonymisation et de partage d'information ;
– et par extension l'ensemble des outils de gestion de la données ou d'information (GED, KM…).
EIM regroupe donc l'ensemble **des processus, technologies et organisations nécessaires pour transformer la donnée en information, l'information en connaissance, et la connaissance en actions** générant un profit pour l'entreprise.

Entité – Une **entité** est une **représentation d'un ensemble d'objets de même nature, concrets ou abstraits.**

ERP (*Enterprise Resource Planning* ou PGI : progiciel de gestion intégré) – Désigne un progiciel comprenant **divers modules qui permettent à une entreprise de gérer d'importantes parties de ses affaires** : planification de la production, achats de produits intermédiaires, gestion du stock de pièces de rechange, relations avec les fournisseurs, services aux clients, suivi de l'exécution des commandes, et aussi comptabilité et gestion des ressources humaines.

ESB (*Enterprise Service Bus*) – Bus applicatif permettant de router de manière sécurisée l'appel à des Services Web. Il s'agit d'un composant clé dans une suite SOA, où des systèmes et des applications nombreux et variés doivent se connecter à un bus et peuvent poster et recevoir des messages/évènements depuis ce dernier.

État de la donnée – On peut distinguer (voir aussi cycle de vie) :
– État métier : le passage d'un état métier à un autre est délimité par un « événement métier ».
– État technique : création, modification, consommation…

ETL (*Extract Transform Loading*) – ETL est un logiciel destiné à extraire des données de diverses sources (bases de données de production, fichiers…), à les transformer et à les charger en général dans une base de données (*data warehouse*, *data*

mart). C'est donc un **outil combinant trois fonctions — extraction, transformation et chargement — pour récupérer les données de bases de données et les déplacer vers d'autres bases**. L'ETL est utilisé pour migrer les données, ainsi que pour convertir des bases de données dans un autre format. La fonction « extraction » prend en charge la lecture des données sources. La fonction « transformation » convertit les données extraites sous une forme permettant de les placer dans d'autres bases de données. La transformation est réalisée grâce à l'utilisation de règles ou de tables d'équivalence ou par combinaison des données extraites avec d'autres données. La fonction « chargement » écrit les données transformées dans la base de données cible.

Exactitude – Notion relative à la qualité des données. Les objets de données représentent-ils bien les **valeurs « du monde réel » des données qu'elles sont censées modéliser** ?

ERM (*Enterprise Rights Management*) – Logiciel de la gestion de droits d'entreprise (ERM) qui **contrôle et impose des politiques d'accès à l'information et à l'utilisation des documents électroniques** au sein d'une entreprise.

Flux – Quantum d'information **nécessaire à l'accomplissement d'une activité, ou produit par une activité**. Exemples : appel d'offre, contrat de fourniture, plan de production… On parle aussi de flux (au sens échanges) entre sous-systèmes fonctionnels ou entre applications (les flux sont caractérisés par des interfaces).

Flux applicatif – Quantum d'information **échangée entre deux applications**.

Flux de données – Représentations techniques des données échangées. Dans certains cas ce sont des messages, dans d'autres ce sont simplement des données représentant une information et qui alimentent un ou plusieurs processus. Le concept de message recouvre pleinement la notion de flux de données, car un message est un événement notifié à un processus avec des données représentant une information. Dans le cas d'un flux simple de données, l'événement est implicite. Quand il s'agit de flux de type message, l'événement est toujours spécifié. Ainsi les flux de données ne servent pas seulement à alimenter les processus, mais également à les synchroniser.

Format canonique ou Format pivot – Format intermédiaire utilisé pour les échanges.

Format de données – Comprend la **description des attributs, le caractère conditionnel ou obligatoire de ces attributs et les règles concernant les valeurs de ces attributs.**
Par exemple : un produit aura comme attribut un prix (attribut obligatoire) dont la valeur est comprise entre 10 et 1 000 euros, le format d'une date peut être JJ/MM/AAAA…

Format pivot – Voir Format canonique : format intermédiaire utilisé pour les échanges.

Front Office – En informatique, le terme ***front office*** est un terme d'architecture logicielle. **Il désigne la partie qui prend en charge l'interface d'une application**, par

opposition au *back office* qui lui regroupe la partie gestion (qui, par rapport à une architecture trois tiers regroupe la partie métier et données).
On retrouve typiquement le ***front office*** dans les sites web commerciaux.

GDR (gestion des données de référence) – Traduction française de **MDM**.

GDS (*Global Data Synchronization*) – Concept qui intervient lorsque **l'information provenant du SI d'un fournisseur de produits et l'information provenant du SI d'un distributeur de produits sont alignées**. Ce concept est principalement utilisé dans le secteur de la grande distribution pour les données Produit. Il est souvent associé à une solution de *Product Information Management* (PIM, un des ancêtres du MDM).

Gestion de métadonnées – Données sur les données, les métadonnées décrivent comment, quand et par qui une série de données a été collectée et la manière dont elles ont été formatées. Les métadonnées sont par exemple indispensables pour comprendre les informations stockées dans les *data warehouses* et sont de plus en plus importantes dans les applications Web utilisant le format XML. **La gestion des métadonnées joue un rôle dans les environnements complexes actuels en permettant de voir immédiatement l'impact des changements de données sur les multiples systèmes interdépendants utilisant ces données.**

Gouvernance – Au sens contemporain, exercice du pouvoir dans une organisation pour atteindre les objectifs fixés. On peut aussi décrire la gouvernance comme **les méthodes pour créer de la valeur et de la performance grâce au système d'information**. Utilise des référentiels (CoBit, ITIL…) promus par des organismes (ISACA, AFAI…) et des outils (BSC, gestion de projet, aide à la décision, qualité de service…).

Gouvernance des données – Ensemble formel des dispositions encadrant les personnes, les processus et les technologies afin de permettre à l'entreprise d'élever la donnée au niveau d'un actif et d'en augmenter la valeur. De manière plus pragmatique, **c'est l'ensemble des processus qui permettent de garantir la qualité, la disponibilité et la sécurité des données**.

GRC (gestion de la relation client) – Voir CRM.

Hiérarchie – Objets organisés selon une série de relations 1 - *n* en cascade. Cette organisation de données est **comparable à un arbre logique, ou chaque membre n'a pas plus d'un père mais un nombre quelconque d'enfants.**

Historisation – Il s'agit de la **sauvegarde des instances** (ou valeurs) d'une donnée.

Hub de données (*data hub*) – Application spécifique qui **réconcilie les données provenant de différents systèmes afin de les mettre à disposition avec une meilleure qualité pour des applications en aval.**

Identifiant – Un identifiant repère de manière unique un objet ou une information dans un système. C'est un **attribut spécifique qui sert à repérer la donnée.** Par exemple le code Siret sert à identifier un établissement d'une société.

Information – Données agrégées en vue d'une utilisation par un humain (par exemple les résultats de requêtes décisionnelles sont des informations). Élément de connaissance susceptible d'être représenté à l'aide de conventions pour être conservé, traité ou communiqué. Désigne aussi un objet métier décrit par ses attributs et la sémantique associée (et donc compréhensible par un humain).

Intégration d'application – Processus consistant à créer une **communication entre des applications ayant des fonctions différentes** comme la GRC, la facturation, la logistique… **mais partageant et modifiant des données communes** comme la liste des clients, le catalogue des produits et des services…

Intégration de données – Processus consistant à **associer deux séries de données ou plus, pour les partager et les analyser** dans un environnement de gestion d'informations commun à toute l'entreprise.

Interface – Mise en forme des données permettant leur passage d'une étape à l'autre d'un traitement. Les interfaces décrivent la structure des informations échangées entre sous-systèmes fonctionnels ou des données échangées entre applications.

Intermédiation – Logiciel intermédiaire (encore appelé *middleware*) entre deux applications ou deux briques applicatives.

ISO (*International Organization for Standardization*) – Organisation internationale de normalisation (http://www.iso.org).

IT (*Information Technology*) – Technologies de l'information. Terme générique qui recouvre les concepts de **gestion de l'information sous toutes ses formes par l'informatique.**

Journalisation – Conservation de l'ensemble des actions effectuées **dans un fichier de trace journalier**. Ce journal peut être archivé afin de reconstituer ultérieurement un état ou une action (auditabilité). Une solution de gestion de données de référence supporte la dernière version de la donnée. La dernière action peut être enregistrée dans des métadonnées de contexte (qui, quand, action…).

LDAP (*Lightweight Directory Access Protocol*) – LDAP est **un protocole permettant l'interrogation et la modification des services d'annuaire**. Un annuaire LDAP respecte généralement le modèle X.500 édicté par l'UIT-T : c'est une structure arborescente dont chacun des nœuds est constitué d'attributs associés à leurs valeurs.

***Mapping* –** Le *mapping* est la **mise en correspondance de modèles de données**. Par exemple, un attribut point de mesure d'un objet métier technique dans un logiciel de MDM peut être traduit par une adresse dans un logiciel de CRM et une localisation dans un autre progiciel métier.

MDM – MDM (*Master Data Management*), encore appelé « référentiel » ou « solution référentielle » par abus de langage, représente une **suite de solutions logicielles permettant la gestion des données destinées à définir un point de vérité unique**. Elle regroupe l'ensemble des données dites de base (ou maître ou *master*

data) en sein d'un référentiel. Celui-ci servira alors de modèle lors de la mise à jour de tel ou tel système ou base de données.

Métadonnées (*metadata*) – De manière synthétique, **donnée qui décrit une donnée**. Les métadonnées sont donc des informations qui renseignent sur la nature de certaines données. Les métadonnées que l'on peut par exemple associer à un document sont : son titre, son auteur, sa date de création… Dans le cadre du décisionnel, elles constituent une sorte de dictionnaire sur lequel le système s'appuie pour comprendre des données utilisées par les différentes applications qui alimentent le *data warehouse*. Les intitulés « Client » d'un PGI et « nom » d'une application comptable peuvent contenir les mêmes informations mais le système ne peut le savoir que si un dictionnaire a été conçu pour lui indiquer qu'il s'agit de la même nature d'informations. Les métadonnées englobent également l'ensemble des informations relatives à la provenance, à l'historique et aux traitements associés aux données.

***Middleware* –** Le *middleware* désigne les logiciels servant **d'intermédiaire entre d'autres logiciels**. On utilise généralement du *middleware* comme intermédiaire de communication entre des applications complexes, distribuées sur un réseau informatique. Par extension, désigne tout logiciel d'intermédiation.

Migration de données – Processus qui consiste à **traduire les données et à les transférer d'un système à un autre avec ou sans transformation de format**. La migration de données est nécessaire lorsque l'entreprise décide d'utiliser de nouveaux systèmes informatiques ou de nouveaux systèmes de gestion de bases de données incompatibles avec les systèmes en place. Elle est généralement réalisée par un ensemble de programmes personnalisés ou de scripts qui transfèrent automatiquement les données.

Modèle conceptuel de données (MCD) – Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. **C'est une représentation graphique des données et des liens qui existent entre chacune d'elles**.
Les concepts de base sont : entité (ou objet ou classe), relation ou association, propriétés ou attributs, identifiant.
On peut utiliser une modélisation UML ou Merise.

Modèle de données – Type de métadonnées qui permet de décrire les différents attributs de la donnée. Il est composé d'un ensemble d'entités, d'associations, de données (propriétés) et de contraintes qui rend compte d'un sujet donné. C'est donc une collection des descriptions de structure de données et des champs (attributs) qui y sont contenus.

Modèle logique de données (MLD) – Modèle qui **décrit la structure de données utilisée sans faire référence à un langage de programmation**. Il s'agit donc de préciser le type de données utilisées lors des traitements. Le modèle logique est dépendant du type de base de données utilisé.
Ainsi, dans un SGBD relationnel, chaque classe d'entité du modèle conceptuel devient une table dans le modèle logique. Les identifiants de la classe d'entité sont parfois appelés clés de la table, tandis que les attributs standard deviennent des attributs de la table, c'est-à-dire des colonnes.

Modèle physique de données (MPD) – Modèle relatif à la conception des bases de données permettant de définir la mise en œuvre de structures physiques et de requêtes portant sur des données. Le MPD, contrairement au modèle logique (MLD) ou conceptuel (MCD) dépend de la base de données et des détails de l'implémentation. Il est dépendant de la plate-forme.
Cette étape consiste donc à implémenter le modèle dans le SGBD, c'est-à-dire à le traduire dans un langage de définition de données. Le langage généralement utilisé pour ce type d'opération est le SQL et plus spécialement le langage de définition de données du SQL.
En pratique, on confond souvent modèle logique et physique, et on fait référence à un seul modèle !

Nettoyage de données – Processus visant à homogénéiser les données pour les rendre plus exploitables. **Le nettoyage des données assure leur intégrité en éliminant les doublons, en corrigeant l'orthographe et en supprimant ou complétant les champs non renseignés.** Les opérations de nettoyage peuvent également couvrir le filtrage, l'agrégation, la vérification de relations…

Nomenclature – Table de valeurs codifiées. Exemple : nomenclature des activités françaises (NAF).

OASIS – Organisation internationale qui standardise la SOA (http://www.oasis-open.org).

Objet métier – Ce sur quoi porte une activité : l'objet métier matérialise le résultat du travail effectué par un acteur de l'entreprise. Structure de données conçue pour représenter les processus et les connaissances d'un métier en particulier. Il s'agit d'un concept défini par un acteur métier.

***Open Group* – Consortium technologique international** destiné à améliorer l'efficacité de travail en jouant un rôle d'intermédiaire entre les acheteurs et les fournisseurs informatiques. L'objectif est de faire gagner du temps et de réduire les coûts ainsi que les risques associés à l'intégration d'une nouvelle technologie au sein d'une entreprise.
Avec sa méthodologie de certification éprouvée et son expertise dans les tests de conformité, l'*Open Group* est un facilitateur international qui permet de fournir l'interopérabilité que les organisations nécessitent pour assurer leur indépendance.
On notera principalement sa méthodologie **TOGAF**.

Pattern – Mot anglais souvent utilisé pour désigner un modèle, une structure, un motif, un type… C'est une description d'une solution réutilisable en réponse à un problème spécifique dans l'ingénierie logicielle (on évoque des patterns d'architecture, mais aussi de développements). Le concepteur d'un **pattern** propose une solution générique pour un problème qui est souvent rencontré dans divers développements. En génie logiciel, un **patron de conception** (*design pattern* en anglais) est un concept destiné à résoudre les problèmes récurrents suivant le paradigme objet.

PDM (*Product Data Management*) – Dénommé en français **GDT (gestion des données techniques)**, système se chargeant uniquement de la gestion des informa-

tions techniques associées au produit (fichiers CAO, plans, documentation…) lors de la phase de conception.

PGI – Voir ERP.

PIM (*Product Information Management*) – Combinaison des technologies, processus et services nécessaires pour **créer et maintenir une description exacte, complète et à jour du produit** quels que soient les canaux de distribution de la donnée, l'organisation utilisatrice, et cela dans un contexte où plusieurs sources de données Produit coexistent. La solution de PIM est souvent considérée comme un préalable pour rendre possible la *Global Data Synchronization* (GDS).

Plan d'urbanisme – Plan qui décrit le **système d'information cible,** ainsi que la trajectoire à suivre pour atteindre cette cible.

PLM (*Product Life Management*) – Représente une suite de solutions logicielles ayant pour vocation **la gestion des données et des processus relative à un produit.** On y trouve des fonctions de collaboration pour la conception d'un produit, tout ce qui concerne son développement, ainsi que le contrôle qualité.

Point d'acquisition – Application qui permet **d'acquérir la donnée (par saisie ou transfert).**

Point de vérité (ou source de vérité, ou point unique de vérité) – Application qui permet de « transformer » la donnée dans **un état à partir duquel elle est considérée comme valide.**

Processus – Ensemble d'activités coordonnées dans le temps dont le résultat satisfait une demande clairement identifiée, interne ou externe à l'entreprise. Le processus est qualifié de transverse lorsqu'il réunit des activités de métiers différents. Exemple : répondre à un appel d'offres.
Définition ISO 9000 : ensemble d'activités corrélées ou interactives qui transforme des éléments d'entrée en éléments de sortie (ISO 9000).

Processus métier lié à la donnée – Processus qui **couvre un ou plusieurs états de la donnée**. Plusieurs processus métier peuvent agir sur un même état de la donnée ou consommer cette donnée.

Processus référentiel – Ce sont les processus **spécifiques à la création, à la modification ou l'accès d'une donnée de référence.**

Profilage – Voir *data profiling***.**

Propriétaire d'informations – Rôle défini dans certains plans sécurité du SI. **Il assure la maîtrise d'ouvrage de la sécurité des données dans son périmètre.** Un propriétaire d'informations est un collaborateur d'une entreprise, formellement désigné par la direction de son entité pour jouer ce rôle sur un périmètre métier fixé. Les responsables de processus-métier sont légitimes pour jouer ce rôle. Sur ce périmètre, les processus métier induisent la constitution d'un ensemble de données et leur traitement.
Le propriétaire d'informations met en œuvre une démarche de classification des informations traitées par le SI qui vise à identifier synthétiquement les besoins de

sécurité d'une manière qualitative et quantitative. Elle aborde en particulier les thèmes suivants :
– Identification des données et traitements à soumettre à la démarche de classification.
– Définition des conditions d'utilisation et d'accès aux données ayant une incidence sur le niveau de vulnérabilité des informations.

Propriétaire des données – Le propriétaire (*data owner*) définit les principales règles relatives à la création, modification et utilisation de la donnée (modèles et instances). Il doit aussi piloter les éléments relatifs à la sécurité (droits d'accès, droits d'utilisation, sensibilité des données…) et à la qualité (mais limités à la définition du niveau de qualité requis pour les performances du métier).

Qualité des données (*data quality*) – Conformité structurelle des données à l´utilisation qu´on souhaite en faire. Améliorer la qualité peut consister par exemple en la correction des occurrences multiples d´un même objet ou le renseignement de champs vides.

Référentiel – Un référentiel (*repository*) s'apparente à une base dans laquelle l'entreprise documente une partie de ses règles de fonctionnement, techniques ou fonctionnelles. Ces informations peuvent décrire comment est structuré le système d'information, selon quelles règles les données de l'entreprise sont transférées et transformées d'une base à une autre, comment se décomposent les principaux processus… Les référentiels contiennent les objets qui représentent les organes de l'entreprise et de son environnement ainsi que leurs caractéristiques. Les gisements de données opérationnelles enregistrent les changements d'états de ces objets au cours de chaque opération effectuée par l'entreprise.
Désigne aussi un ensemble structuré d'information, utilisé pour l'exécution d'un logiciel et constituant un cadre commun à plusieurs applications. Contient des informations non structurées (par opposition à *registry* qui ne contient que des informations structurées).
Désigne aussi **par abus de langage une application de type MDM**.

Registry – Annuaire d'objets informatiques ne contenant que des informations structurées. Exemple : *registry* UDDI pour un annuaire de services.

Repository – Annuaire d'objets informatiques contenant des descriptions complémentaires non structurées. Exemple : *repository* de services qui décrit de façon exhaustive les services disponibles, y compris avec description textuelle.
Par extension désigne un référentiel d'entreprise. Base de données centrale qui stocke et gère l'information d'une entreprise et de ses systèmes dans le but de servir de point de référence dans des phases ultérieures de traitement. On peut même stocker dans un référentiel d'entreprise toutes les composantes d'une application (code source, interfaces graphiques, traitement), de même que l'analyse du système, le traitement des données et la documentation en ligne.

Rôle – Ensemble de fonctions à réaliser (définir le modèle de données, gérer la qualité…). Un acteur tient un ou plusieurs rôles.

Sémantique – Ce **qui est relatif au sens** (d'un mot, d'un texte) ou à une intention (d'une action, d'une organisation).

Service – Fonction rendue par une entité pour le compte d'une autre entité, un composant qui est utilisable par une application **(vocabulaire consacré dans l'architecture SOA : un service est défini par son interface).**

Services universels de données (*Universal Data Services*, UDS) – Une architecture de **services universels de données fournit une fondation technologique pour distribuer des informations fiables, compréhensibles et actualisées provenant de multiples sources complexes.** Basé sur une architecture orientée services, UDS réduit la complexité des systèmes, processus et ressources, et permet de faire aboutir plus rapidement des projets à forte valeur ajoutée, tels que les initiatives de migration et de synchronisation de données, de création de hubs de données, de *data warehousing* et de supervision des activités métier.

SI (système d'information) – Ensemble constitué par la définition des processus des métiers et par celle des flux d'information associés. Un SI comprend **l'ensemble des moyens (organisation, acteurs, procédures, systèmes informatiques) nécessaires au traitement et à l'exploitation des informations** dans le cadre d'objectifs définis au niveau de la stratégie de l'entreprise, des métiers, de la réglementation.

SLA (*Service Level Agreement*) – Contrat définissant les **obligations d'un hébergeur de service vis-à-vis d'un fournisseur de service** en matière de niveau de qualité de service.

SOA (*Service Oriented Architecture*) – Modèle d'architecture applicative mettant en œuvre des connexions en couplage lâche entre divers composants logiciels (ou services). Un service désigne une action exécutée par un composant « fournisseur » à l'attention d'un composant « consommateur », hébergé éventuellement sur un autre système.

Socle de gouvernance – Ensemble de solutions techniques réutilisables permettant de participer à la gouvernance. Par exemple :
– audit des erreurs ;
– gestion des métadonnées ;
– tableau de bord ;
– audit de sécurité.

Socle technique – Ensemble de produits qui permettent de **mettre en place progressivement plusieurs fonctionnalités techniques réutilisables** dans plusieurs projets :
– MDM ;
– *workflow* ;
– intermédiation ;
– alertes (notifications) ;
– sécurité (annuaire LDAP)…

Source de vérité ou point de vérité – Application à partir de laquelle la **donnée est considérée comme valide.**

Sous-système – Sous-ensemble du **système d'information, autonome vis-à-vis de son fonctionnement et de son évolution**. Il constitue un bloc fonctionnel et **est défini par les objets métier qu'il détient.**

Sponsor – Ce rôle est utile à la bonne mise en œuvre de la démarche MDM. Il définit et valide la stratégie de l'entreprise. **Il assure la mise à disposition des moyens (ressources et personnels) et pilote les alignements stratégiques** en cours de déploiement de la démarche.

Structure de donnée – Ensemble organisé de données ayant quelque chose en commun et qu'on a groupées pour leur traitement. Exemple : (nom, prénom, âge, profession).

Synchronisation de données – La synchronisation couvre l'envoi, la réception et la mise à niveau des données entre les différents systèmes. On dit alors que les données sont cohérentes entre applications. Dans les environnements où plusieurs applications utilisent les mêmes données et où un des utilisateurs modifie un des objets partagés, **la modification apportée doit parfois être immédiatement propagée aux autres applications.**

Table de valeur – C'est la **liste des valeurs possibles d'une donnée**. Exemple : liste de segments *marketing* appelée aussi « domaine de valeurs ».

Template (gabarit ou modèle) – Forme de référence à partir de laquelle sont créés des objets qui présentent des caractéristiques communes. Dans un tableur, un logiciel de traitement de texte ou tout autre logiciel d'application (gestion de contenus ou publication par exemple), on retrouve souvent un modèle de document contenant des images, du texte et des éléments de formatage qui sont souvent utilisés pour créer d'autres documents, par un simple ajout d'informations, afin de permettre à l'utilisateur de gagner du temps.

Transcodification – Il s'agit d'une **table de correspondances de valeurs codifiées**. Exemple : codes articles fournisseurs en codes articles clients.

UML (*Unified Modelling Language*) – UML (que l'on peut traduire par « *langage de modélisation unifié* ») est une **notation permettant de modéliser un problème de façon standard.** Ce langage est né de la fusion de plusieurs méthodes et est devenu désormais la référence en termes de modélisation objet, y compris pour les données.

Urbanisme du système d'information – Par analogie avec l'urbanisme d'une ville, cette expression désigne la **modélisation « à grosses mailles » du SI** d'une entreprise avec ses divers domaines, leurs processus et les relations qu'ils entretiennent, les règles qu'un SI doit respecter pour être cohérent et utiliser de façon efficace les ressources partagées par les divers processus.

Versionning – Il s'agit de la **sauvegarde des modèles de données**.

Vue unique (ou unifiée) – Capacité à **fournir à l'ensemble des départements de l'entreprise une vue standardisée, basée sur les mêmes données et définitions de**

données, pour un sujet spécifique. On parle par exemple de vue unique du client. Voir aussi la notion proche de *hub* de données.

W3C (*World Wide Web Consortium*) – Organisation internationale qui standardise la SOA et le Web (http://www.w3.org).

***Web service* ou service Web – Composant applicatif indépendant (appelé « service ») accessible par l'entremise d'une interface bien définie**, qui peut interagir avec des applications en utilisant des protocoles de communication standardisés indépendamment du système d'exploitation et des langages de programmation utilisés.
Les services Web ont recours à un ensemble de standards : le langage XML pour décrire les informations, la norme UDDI pour trouver les services dont on a besoin, le langage WSDL pour décrire leur interface et le protocole SOAP pour les exécuter à distance.

***Workflow* – Logiciel qui permet d'organiser, de faire fonctionner et de contrôler un processus**. Le *workflow* comporte la définition des masques des documents échangés entre les acteurs du processus et les règles qui codifient son fonctionnement : programmation des routages, délais, alarmes, compteurs de délais et de volumes, édition de comptes rendus automatiques. Il fait intervenir des actions De manière plus simple, on parlera de *workflow* de saisie lorsque plusieurs personnes doivent coopérer pour créer une donnée complète validée.

WS-I (*Web Services Interoperability*) – Organisation internationale qui **standardise la SOA et en particulier les services Web**.

XML (*eXtensible Markup Language*) – Langage de **description de balises permettant de décrire une structure de données insérée dans un document, un fichier.** Permet la structuration, le typage de champs et la standardisation de l'accès aux données par l'intermédiaire d'un XML Schema.
Sert aussi à décrire des formats de données. C'est donc un langage de codage de données dont l'objectif est, dans un échange entre systèmes informatiques, de transférer, en même temps, des données et leurs structures. Permettant de coder n'importe quel type de donnée, depuis l'échange EDI jusqu'aux documents les plus complexes, son potentiel est de devenir le standard universel et multilingue d'échange d'informations.

XML Schema – C'est un **langage de description de format de document XML permettant de définir la structure d'un document XML**. Un schéma XML est lui-même un fichier XML. La connaissance de la structure d'un document XML permet notamment de vérifier la validité de ce document. Un fichier de description de structure (*XML Schema Description* ou fichier XSD) est donc lui-même un document XML.

XSD (*XML Schema Definition*) – XSD **définit de façon structurée le type de contenu, la syntaxe et la sémantique d'un document XML**. Il est également utilisé pour valider un document XML, c'est-à-dire vérifier si le document XML respecte les règles décrites dans le document XSD.

Bibliographie

[Berson2007] Alex BERSON, Larry DUBOV, *Master Data Management and Customer Data Integration for a Global Enterprise*, Mac Graw Hill Osborne, 2007.

[Rivard2008] François RIVARD, Georges ABOU HARB, Philippe MERET, *Le système d'information transverse*, Hermes-Lavoisier, 2008.

[Bonnet2007] Pierre BONNET, Jean-Michel DETAVERNIER, Dominique VAUQUIER, *Le système d'information durable*, Hermes-Lavoisier, 2007.

[Abou Harb2002] Georges ABOU HARB, François RIVARD, *L'EAI par la pratique*, Eyrolles, 2002.

[Rivard2003] François RIVARD, Thomas PLANTAIN, *L'EAI au service de l'entreprise évolutive*, Maxima, 2003.

[The Data Warehouse Institute2002] *Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data*, The Data Warehouse Institute, 2002.

[ISO2005] *Systèmes de management de la qualité – Principes essentiels et vocabulaire*, ISO, 2005.

[Karel2006] Rob KAREL, Christopher MINES, R "Ray" WANG, Kyle McNABB, Jamie BARNETT, *Introducing Master Data Management*, Forrester Research, 2006.

[Lefébure2005] Réné LEFÉBURE, Gilles VENTURI, *Gestion de la relation client*, Eyrolles, 2005.

[White2007] Andrew WHITE, Mark A. BEYER, *The Role of Metadata in Master Data Management*, Gartner, 2007.

[Davenport1992] Thomas H. DAVENPORT, Robert G. ECCLES, Laurence PRUSAK, *Information Politics* — Sloan Management Review, 2007.

Index

INFOPRO

TYPE D'OUVRAGE

L'ESSENTIEL	SE FORMER	RETOURS D'EXPÉRIENCE

Franck Régnier-Pécastaing
Michel Gabassi
Jacques Finet

- **MANAGEMENT DES SYSTÈMES D'INFORMATION**
- APPLICATIONS MÉTIERS
- ÉTUDES, DÉVELOPPEMENT, INTÉGRATION
- EXPLOITATION ET ADMINISTRATION
- RÉSEAUX & TÉLÉCOMS

MDM

Enjeux et méthodes de la gestion des données

Cet ouvrage s'adresse à tous les responsables, tant côté maîtrise d'ouvrage que maîtrise d'œuvre, et à tous les dirigeants, urbanistes, chefs de projet… qui cherchent à améliorer la valeur de l'information détenue et utilisée par l'entreprise.

Comment optimiser l'interaction et la synchronisation de données du SI et assurer leur qualité (cohérence, mise à jour, absence de doublon...) ? Comment transformer une vision stratégique liée à la donnée en une réalité ?

Ce livre répond à ces interrogations en exposant des méthodes et des solutions de gestion des données de référence et met l'accent, en particulier, sur **les données de référence** et sur la notion essentielle de « **point de vérité** ».

La première partie expose les **concepts, besoins et enjeux** de la gestion de données.

La deuxième partie présente les **bonnes pratiques, les architectures et les solutions** pour améliorer cette gestion. Le **MDM (*Master Data Management*)** y est en particulier détaillé.

La dernière partie propose des **méthodes et des organisations** s'appuyant sur le concept clé de **gouvernance des données**.

FRANCK RÉGNIER-PÉCASTAING
est responsable des offres Entreprise Information & Master Data Management chez Logica Management Consulting.

MICHEL GABASSI
est ingénieur architecte à la Direction Informatique et Télécommunications d'EDF/GDF Suez et responsable du catalogue des solutions pour le middleware et les progiciels.

JACQUES FINET
est ingénieur urbaniste à la Direction Informatique et Télécommunications d'EDF/GDF Suez.

SUPPLÉMENTS en ligne

logica management consulting
www.yourpotential.tv

ISBN 978-2-10-053555-2

www.dunod.com

DUNOD