

architecture et urbanisation des SI

1. Introduction

Le Système d'Information (SI) est aujourd'hui un élément central du fonctionnement d'une organisation. Un Système d'Information peut être défini comme un **ensemble de ressources** (personnel, logiciels, processus, données, matériels, équipements informatique et de télécommunication...) permettant la collecte, le stockage, la structuration, la modélisation, la gestion, la manipulation, l'analyse, le transport, l'échange et la diffusion des informations (textes, images, sons, vidéo...) au sein d'une organisation. Exemples de ressources informatiques : fichiers de données, bases de données et SGBD (Système de Gestion de Bases de Données), progiciels intégrés (ERP, ...), outils de gestion : gestion clients (CRM : Customer Relationship Management), gestion de la chaîne logistique (SCM : Supply Chain Management), gestion des employés (ERM : Employee Relationship Management), outils de travail collaboratif (GroupWare), applications métier, serveurs d'application, serveur de présentation (Web,...), système de Workflow, architecture d'intégration (EAI : Enterprise Architecture Integration, SOA : architectures orientées services), infrastructure réseau, ...

La définition donnée précédemment laisse entrevoir la complexité du SI dont les déclinaisons vont s'exprimer à l'aide

de différentes architectures. Il est alors primordial aujourd'hui de différencier système d'information (SI) et système informatique. Un SI peut être considéré comme une vue « automatisable » des métiers d'une organisation et une vue fonctionnelle de l'informatique, donc indépendante de l'implémentation technique (figure 1). Le SI est plus pérenne que l'architecture informatique. Les évolutions applicatives et techniques peuvent être indépendantes du SI en raison de l'évolution des technologies, des configurations ou des besoins utilisateurs.



Figure 1. Découplage : Processus métier / Système d'information / Informatique (applications + architecture technique)

La conception de SI d'une entreprise requière des méthodes d'analyse de l'entreprise afin de modéliser les informations et les données, les flux d'information échangés ainsi que les traitements à appliquer sur ces données. Ces traitements sont identifiés grâce à l'analyse des processus métier (Figure 2).

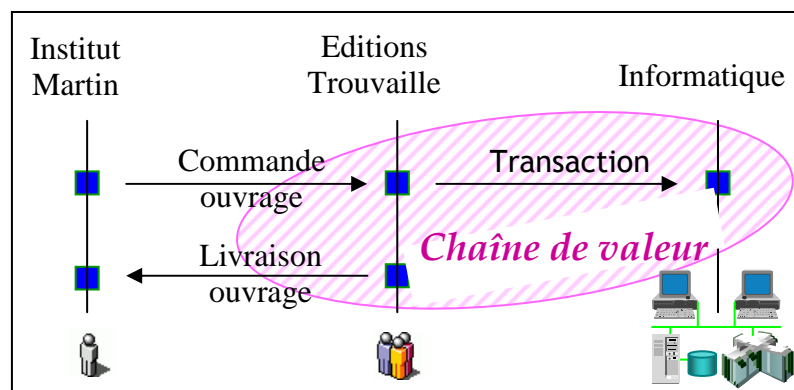


Figure 2. Exemple de diagramme de workflow d'un processus métier général

Des modèles ou langages de modélisation sont donc nécessaires. La méthode Merise et plus récemment UML sont les plus utilisés notamment en France dans la conception de SI.

Dans beaucoup d'organisations, il ne s'agit plus aujourd'hui de concevoir un système d'information mais de le faire évoluer au rythme des besoins tout en exploitant les avancées technologiques.

Force est de constater que la complexité des SI est proportionnelle à la complexité croissante des technologies et des organisations elles-mêmes. Le SI doit répondre aux enjeux stratégiques de l'entreprise, au développement du marché, supporter l'évolution des métiers et des fonctions au sein de l'organisation, et également supporter l'évolution du périmètre de l'organisation (fusion, intégration de différentes entreprises). Le SI doit donc être évolutif, réactif, flexible, ouvert et également sécurisé. C'est l'objectif de la démarche d'urbanisation des systèmes d'information.

2. Conception

Plusieurs méthodes de conception de SI co-existent et sont exploitées différemment selon les pays. Parmi celles-ci, on peut citer Merise, UML, AXIAL, IDEF...

2.1. Merise

La méthode Merise est née à la fin des années 70 en France avec pour objectif de définir une démarche de conception de SI.

Le principe de base de la méthode Merise repose sur la séparation des données et des traitements. L'organisation des données semble plus pérenne que la définition des traitements qui évolue en fonction de l'évolution des métiers, des fonctions et des utilisateurs. La méthode Merise intègre trois dimensions appelée cycles : le cycle d'abstraction, le cycle de vie et le cycle de décision. Le **cycle de vie** décrit les phases du projet de

construction du SI du schéma directeur à la réalisation. La **dimension décisionnelle** décrit des phases de validation du projet de construction du SI en impliquant la majorité des acteurs ou utilisateurs du SI afin de s’assurer de leur adhésion au futur SI au sein de l’organisation.

Le **cycle d’abstraction** se décompose en 3 couches (Cf. Figure 3), chaque couche correspondant à une modélisation (Données et Traitement) du système d’information.

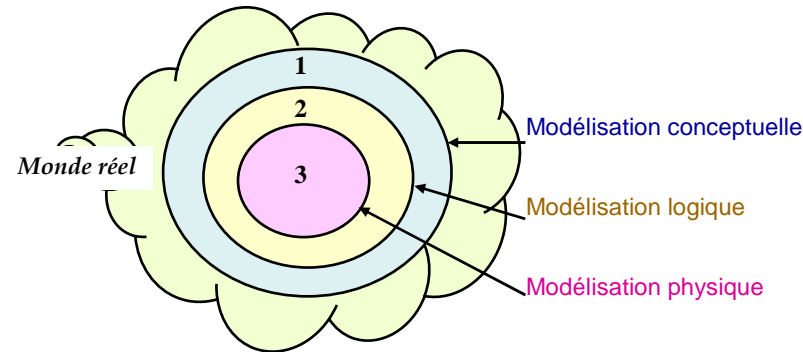


Figure 3. Cycle d’abstraction de Merise

Le **cycle d’abstraction** a pour objectif, à partir de l’expression des besoins, de répondre au QQQQC à savoir : Quoi, Qui, Où, Quand, Comment, concernant les données et les traitements.

	Données	Traitements	
<i>Quoi</i>	Quelles informations utiles ? (MCD)	Pour faire quoi ? (MCT)	Conceptuel
<i>Qui, Où, Quand</i>	Quelle structure de données ? (MLD)	Qui fait quoi et où, et quand ? (MOT)	Logique
<i>Comment</i>	Comment stocker les données ? (MPD)	Comment fait-on ? (MOpT)	Physique

Figure 4. Phases d’analyse et de conception des données et traitements de Merise

Les modèles décrivant les données sont (Figure 4) : le Modèle Conceptuel de Données (MCD), le Modèle Logique de Données (MLD) et en fin le Modèle Physique de Données (MPD). Concernant les traitements, le découpage est symétrique avec le Modèle Conceptuel de Traitement (MCT),

le modèle Organisationnel de Traitement (MOT) et enfin le modèle Opérationnel de Traitement (MOpT). Le niveau d'abstraction décroît au fil des modèles c'est-à-dire que le modèle conceptuel se veut être proche de la représentation réelle (vue utilisateur) et le modèle physique, proche de la représentation informatisée ou implémentée.

Couche 1 : la Modélisation Conceptuelle des données (MCD) a pour objectif de décrire le monde réel sous la forme d'entités et de relations entre ces entités : modèle entité-association (Figure 5a). Ces entités sont appelées Classes dans le langage orienté objet UML (Figure 5b).

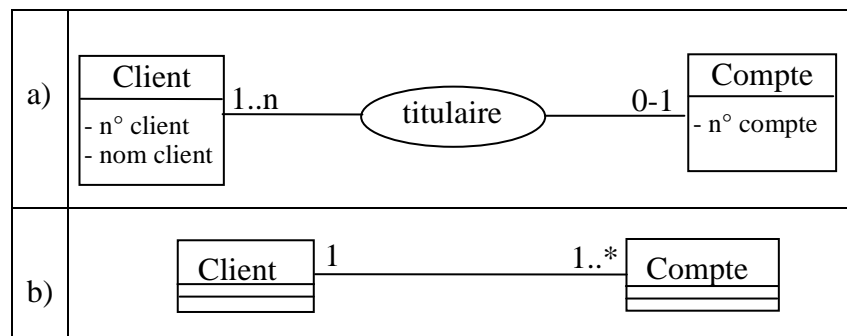


Figure 5. Modélisation conceptuelle de données

La modélisation conceptuelle des traitements (MCT) a pour objectif de décrire des processus métier en interaction avec l'extérieur de type : un événement déclenchant provoque une transformation du système d'information pour produire un résultat. Les flux d'échange sont analysés (Figure 6). L'enchaînement des différentes opérations est ensuite décrit (Figure 7).

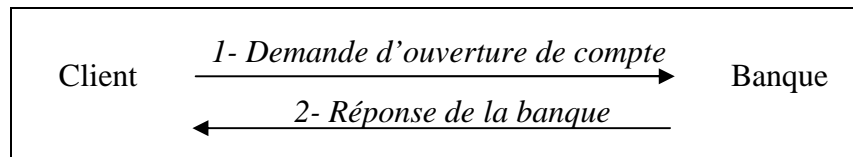


Figure 6 : Diagramme de flux pour la modélisation conceptuelle de traitement

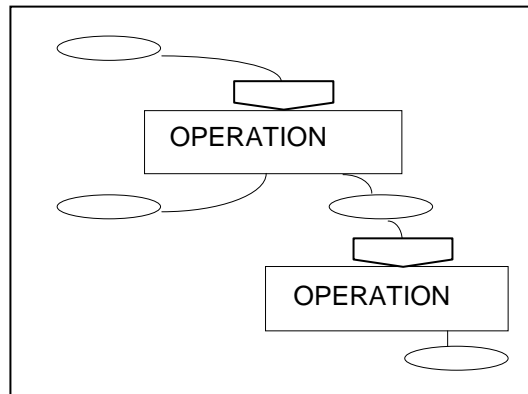


Figure 7 : Modèle conceptuel de traitement

Couche 2 : la Modélisation Logique exprime un choix de structuration pour les données et les traitements. Il s'agira par de décrire les données dans la structure de données choisi : tables de la base de données. Par exemple, l'entité Client est transformée en une table de base de données appelée Client dont les attributs sont détaillés avec déclaration des identifiants uniques appelés clés. Par exemple, (Figure 8) l'identifiant unique d'un client est son numéro. A un numéro de client ne correspond qu'un seul client.

Table Client	<u>N°client</u>	Nom client
--------------	-----------------	------------

Figure 8. Modèle logique de données

Concernant les traitements, le modèle organisationnel de traitement précise le MCT en détaillant notamment les opérations redécoupées en « procédures fonctionnelles ».

Couche 3 : la Modélisation Physique présente le modèle d'implémentation à savoir le choix de matériel informatique (logiciel, outil, système d'exploitation, machine) pour le

système d'information en termes de support de données et de traitements (produit de SGBD, par exemple Oracle, langage et environnement de développement...). La description des données est réalisée dans le langage de définition de données du produit logiciel choisi (ex : en SQL pour Oracle). Les types ou formats de données sont décrits à ce niveau. Par exemple, l'entité Client correspondra à une table de base de donnée et le numéro de client, appelé attribut de l'entité client et noté : n°client (Figure 4a), pourra être déclaré comme une valeur entière et le nom du client comme valeur chaîne de caractères c'est-à-dire un mot. Au niveau traitement, les procédures voire programmes sont détaillés.

2.2. UML : Unified Modeling Language, norme OMG (Object Management Group)

UML que l'on peut traduire en français comme langage de modélisation objet unifié est un langage de description orienté objet qui permet de modéliser une application selon une vision objet. Un objet est décrit par les attributs qui le compose et les traitements appelés méthodes qui peuvent lui être appliqués. Par exemple, l'objet client possède un numéro et un nom. Les méthodes applicables à l'objet client peuvent être : consulter le client, créer, modifier ou supprimer un objet client. UML se compose d'un ensemble de diagrammes (Figure 9) dont certains ont leur équivalent en Merise (diagrammes : d'organisation, d'objets, de classes, de composants, de déploiement, d'utilisation, de collaboration, de séquences...) qui peuvent être exploités pour décrire un système d'information.

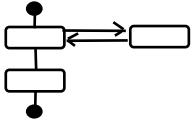
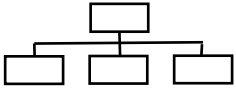
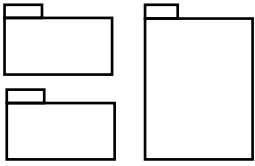
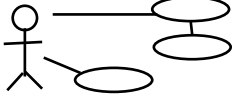
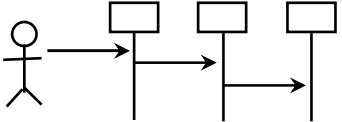
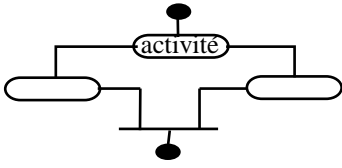
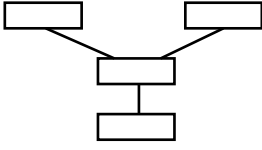
Diagrammes UML	d) 
a) 	e) 
b) 	f) 
c) 	g) 

Figure 9. Quelques Diagrammes UML exploitables pour la conception de SI

a) Diagramme d'organisation, b) diagramme de cas d'utilisation (Use Case), c) diagramme d'activité, d) diagramme d'état, e) diagramme de classes, f) diagramme de séquences, g) diagramme de collaboration.

3. Architectures

Si le concept d'architecture de SI n'est pas récent, il n'en reste pas moins compliqué. En effet, l'architecture d'un SI a de multiples représentations et il serait plus approprié de parler d'architectures de SI au pluriel. Concernant les méthodes de conception d'architecture de SI, aucune méthodologie n'a réussi à s'imposer contrairement à UML pour la conception d'architectures logicielles. Toutefois, les diagrammes UML peuvent être exploités lors de la conception ou reconception de système d'information d'où parfois la confusion possible entre architecture SI et architecture logicielle. Le tableau ci-après

exprime quelques différences entre les deux types d'architectures tant en termes de concepts que de composants.

Architecture de SI	Architecture logicielle
Blocs fonctionnels, référentiels de données, flux de données	Module logiciel, composant, classe
Des applications	Une application
Processus métier, activités	Spécifications fonctionnelles
Spécifications techniques de l'ensemble du SI (ensemble des composants et modèles de données)	Spécification des classes logicielles

Figure 10. Architecture de SI vs Architecture logicielle

Dans le domaine des Systèmes d'Information, de nombreux types d'architectures existent. Ces architectures sont parfaitement identifiées et sont issues du résultat des différentes phases de conception d'un Système d'information. Quelques exemples sont détaillés ci-après :

- Architecture fonctionnelle : représentation des fonctions issues de l'analyse des processus métier. *Exemple : Gestion des Comptes.*
- Architecture applicative : représentation des applicatifs (composants logiciels) et des flux échangés et définition de l'implantation sur l'architecture technique
- Modèle/Architecture logique : représentation virtuelle d'une architecture, abordable aux interlocuteurs. On peut aussi parfois parler d'*architecture applicative logique*.

- Architecture technique : représentation des techniques et standards de construction : Systèmes d'exploitation, SGBD, serveurs, middleware, types de réseau...

On distingue également l'architecture d'implémentation, d'exécution, de déploiement, l'architecture physique...

4. Urbanisation

Urbanisation rime avec simplification et intégration afin de répondre aux enjeux stratégiques, organisationnels, et technologiques de l'entreprise. La démarche d'urbanisation de système d'information est née dans les années 80 au sein des banques. En effet, à partir des années 60, les systèmes d'information se sont construits au fil de l'eau par ajout successifs d'applicatifs et de structures de données sans souci de cohérence globale. Les propositions d'évolution au sein de l'architecture venaient souvent de la direction informatique indépendamment de l'évolution stratégique de l'entreprise. Les années 80 ont donc vu naître les architectures complexes dites

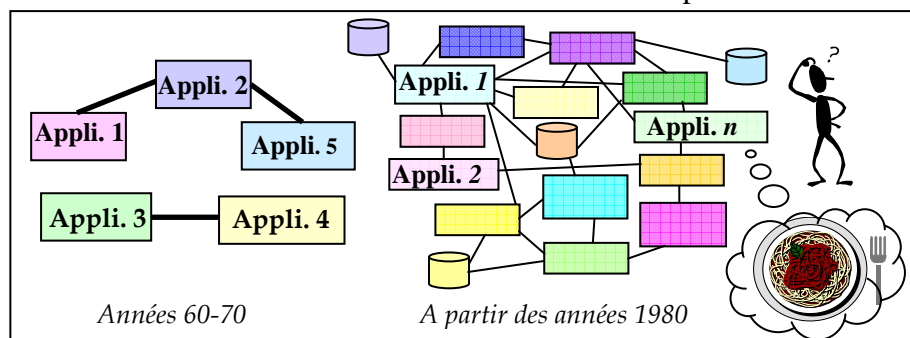


Figure 11. Cartographie d'architectures de systèmes d'information

Il est alors utopique de vouloir tout reconstruire et comme dans une ville, le système d'information doit être en mesure de supporter des évolutions et réorganisations permanentes. Le défi de l'urbanisation est de construire une architecture de SI flexible. L'enjeu est alors le suivant : faire évoluer le système voire refaire certains morceaux, sans détruire l'existant, en

intégrant des composants diverses et en exploitant les avancées technologiques dans un souci de cohérence générale, de réactivité et de flexibilité.

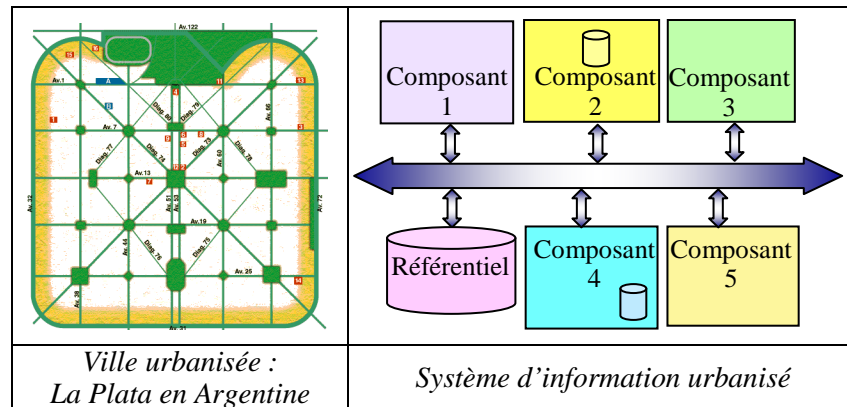


Figure 12. Systèmes urbanisés

4.1. Démarche d'urbanisation

La démarche d'urbanisation recentre le pilotage de l'évolution du système d'information sur la stratégie et les besoins des **métiers** de l'entreprise ou organisation concernée. Elle est basée sur un modèle en quatre couches successives : **Métier**, **Fonctionnelle**, **Applicative** et **Technique**. A partir d'objectifs stratégiques clairement identifiés, les **processus métier** à mettre en oeuvre sont alors identifiés. Puis les **fonctions** et informations utilisées par les processus sont alors détaillées et enfin, les **applications** et **l'architecture technique** permettant d'implémenter ces fonctions sont spécifiées. On peut alors parler d'une démarche « top-down », à savoir démarche de conception descendante. L'urbanisation consiste alors de passer d'un système d'information existant à un SI cible par des étapes successives de description ou construction d'architectures (Figure 13).

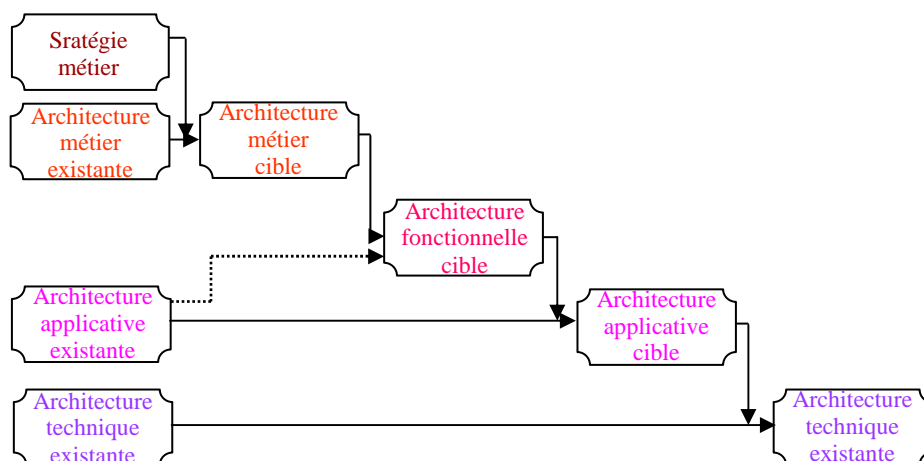


Figure 13. Démarche d'urbanisation top-down

Cette démarche est également exploitée pour la conception d'architecture d'entreprise. Une démarche « bottom-up » (ascendante) peut également être réalisée, poussée par les avancées technologiques et la volonté de réorganisation technologique et applicative, parallèlement ou indépendamment de la démarche « top-down ».

4.2. Cartographies

La cartographie est un outil central de la démarche d'urbanisation. Chaque architecture de la démarche d'urbanisation peut être décrite par une cartographie représentant son POS : plan d'occupation du sol (Figure 14).

Architecture	Cartographie	Eléments représentés sur la cartographie
Architecture métier	Carto Processus	Objectifs stratégiques Processus / Tâches
Architecture fonctionnelle	Carto Fonctionnelle	Système d'information Domaines fonctionnels Référentiels et flux
Architecture applicative	Carto Applicative	Systèmes informatiques Applicatifs/Progiciels Composants, Objets Métiers, flux...
Architecture technique	Carto technique	Matériels (serveurs, poste de travail...) Logiciels (Système d'exploitation, SGBD...) Architecture réseau

Figure 14. Cartographies

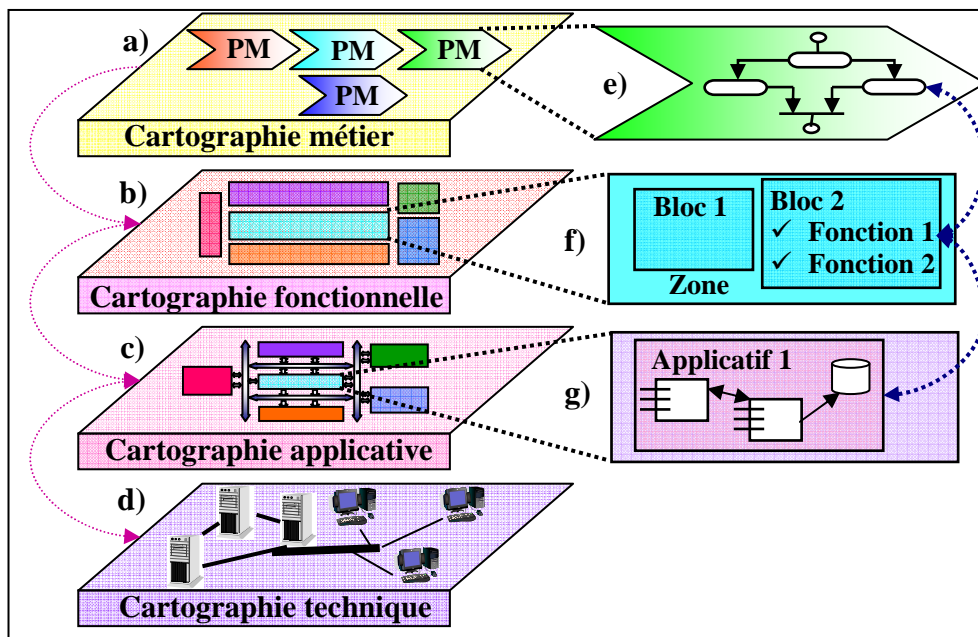


Figure 15. Cartographies détaillées des architectures SI

4.2.1 Cartographie de l'architecture Métier (Figure 15 a)

La cartographie métier représente une vue de l'architecture métier par une formalisation du métier de l'entreprise en répondant aux questions quoi, qui, où et pourquoi (cf. Méthode Merise). L'objectif est de définir les besoins du métier vis-à-vis du système d'information. La cartographie Métier ou cartographie processus décrit les processus métier (PM) de l'entreprise sous forme de diagrammes (Figure 15a et 16). Chaque processus peut ensuite être détaillé en présentant l'enchaînement des activités à l'aide de diagrammes de workflow ou en exploitant les diagrammes UML (Use Case et diagramme d'activité Figure 15e). La description des activités implique donc la spécification des OM : objets métiers (quoi), des entités organisationnelles (qui) et des processus qui répondent à des objectifs (pourquoi). Enfin, la cartographie Métier de l'existant peut être associée à une cartographie

Métier cible si des évolutions métier sont envisagées en réponse à de nouveaux objectifs stratégiques.

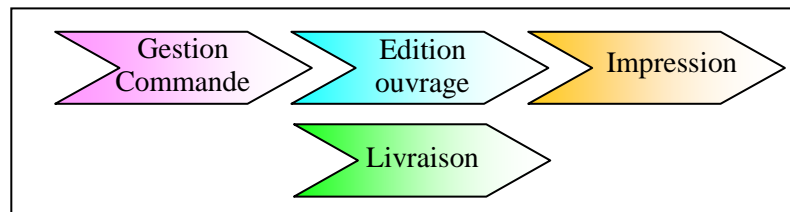


Figure 16. Exemple de cartographie de processus métier

4.2.2 Cartographie de l'architecture Fonctionnelle (Figure 15b)

La cartographie fonctionnelle décrit les fonctions mises en œuvre pour réaliser les activités (issues des processus cf. Figure 15e et 15f) décrites dans l'architecture métier. Il s'agit de décrire les fonctions à valeur ajoutée indépendamment de l'implémentation. Le but recherché est une organisation logique des fonctions, fortement découplée et sans redondance. L'architecture fonctionnelle est découpée en Zones, elles-mêmes redécoupées en Quartier puis îlots ou blocs fonctionnels indépendants et autonomes (Figure 15f). L'îlot, qui contient les fonctions est la plus petite entité du découpage. Prenons l'exemple d'une zone fonctionnelle banque commerciale, plusieurs quartier composent cette zone dont le quartier de gestion l'épargne et le quartier de gestion des crédits. Le quartier de gestion de l'épargne peut être redécoupé en différents blocs selon les types de liquidité : comptes chèques, épargne logement (PEL, CEL), épargne liquide (CODEVI...)... Les données manipulées par ces fonctions sont également décrites. En effet, un bloc fonctionnel est seul propriétaire des données qu'il manipule, afin de faciliter une organisation modulaire. Des règles d'urbanisation ont été définies afin d'aider au découpage de l'architecture fonctionnelle. Le découpage consiste premièrement à séparer les zones de

référentiels (données), d'échange (communication avec l'extérieur), de gestion interne (finance, RH, informatique...), de stratégie et règles de décision, et les zones opérationnelles (métier).

4.2.3. Cartographie de l'architecture applicative (Figure 15c)

L'architecture applicative est une vue informatique et dynamique du système d'information décrit dans la couche fonctionnelle. L'objectif est la distribution et la réutilisation des fonctions applicatives.

L'architecture applicative est décrite par la cartographie de l'ensemble des blocs ou composants applicatifs (composants logiciel, progiciels, ...) et de leurs échanges répondant à l'organisation fonctionnelle spécifiée précédemment. Dans l'absolu, un bloc fonctionnel devrait correspondre à un bloc applicatif. L'existant et son intégration ne permettent pas toujours de respecter cette contrainte. Un progiciel intégré correspondra à un îlot ou bloc applicatif (élément de granularité la plus faible, non « découparable » Figure 15g).

Les structures de données sont également décrites au niveau de l'architecture applicative. Les accès à ces données ainsi que la gestion de leur persistance (sauvegarde, sécurité) sont également décrite à ce niveau. UML ou Merise offrent les modèles et diagrammes utiles à la description de l'architecture applicative.

4.2.4 Cartographie de l'architecture technique (Figure 15d)

Les infrastructures nécessaires au déploiement des composants définis dans la couche applicative ainsi que la manière dont ils communiquent, sont décrites au niveau de l'architecture technique. Le principal objectif lors de la conception de la

couche technique est la mutualisation des plateformes techniques dans le but de réaliser des économies d'échelle. La modélisation de la couche technique est principalement constituée par des diagrammes qui permettent de montrer la connexion entre les serveurs.

4.3 Urbanisation et architectures d'intégration

Les approches SOA et EAI sont des architectures d'intégration permettant de concrétiser la mise en œuvre de SI urbanisés. Elles peuvent être complémentaires.

SOA : Services Oriented Architecture. La démarche de conception d'architectures orientées services répond aux besoins de l'urbanisation du système d'information par l'approche modulaire en couches qu'elle propose. L'approche SOA concerne l'architecture applicative. Les composants applicatifs publient des services regroupés dans un annuaire. L'orchestration (l'enchaînement) de services permet de réaliser des fonctions requises pour la réalisation d'activités de processus métier. Une décomposition des services en couches de la couche métier (service métier : *exemple « consultation des comptes-client »*) jusqu'à la couche données (service de base au niveau entité : *exemple « consultation table client »*) est organisée. Le déploiement des architectures SOA est favorisé par le développement du web et des Web Services (SOAP, WSDL, XML), des serveurs d'application (J2EE, .NET) et la nécessité croissante de systèmes d'information ouverts et interopérables (communicants vers l'extérieur, collaboration inter-entreprises...).

EAI : Entreprise Architecture Integration. L'objectif d'une solution EAI est de réaliser l'intégration des applications sur une architecture informatique commune afin de leur permettre de réaliser des échanges (utile notamment dans le cadre de fusion d'entreprise). Exemple : permettre, à une application, la

consultation des données d'une autre application ou encore la mise à jour des données d'une autre application.