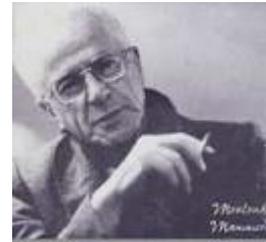


République Algérienne Démocratique et Populaire.
Ministère de L'Enseignement Supérieur et de la recherche
Scientifique.

Université Mouloud Mammeri de Tizi-Ouzou.

Faculté de Génie Electrique et D'informatique.

Département D'informatique.



Mémoire

En vue de l'obtention du diplôme de master en informatique

Option : Systèmes d'Information

Thème

Urbanisation des systèmes d'information

Dirigé par :

M^r Malik Si-Mohamed

Réalisé par :

M^{elle} Kerriche Hadjila

2010-2011

Remerciements

Il me serait impossible de citer nommément toutes les personnes qui m'ont aidée, encouragées et soutenues afin que ce travail voie le jour. Que toutes personnes trouvent ici l'expression de ma sincère reconnaissance.

Je veux tout particulièrement remercier Monsieur Malik Si-Mohamed, pour avoir accepté de diriger mon travail. Son aide, sa disponibilité, et ses conseils, durant toute cette période, m'ont toujours redonné confiance et volonté.

Je remercie vivement tous les enseignants qui ont participé à ma formation primaire, secondaire et supérieure.

A Monsieur Samir HAMEG pour son aide précieuse et sa disponibilité.

Je tiens à remercier également les membres du jury qui m'ont fait l'honneur d'avoir accepté de rapporter et d'examiner ce travail.

Mes remerciements s'adressent également à tous mes amis et tous les étudiants de ma promotion en particulier Lehad Lynda et Hameg Mohamed.

Je n'omettrais pas de remercier ma chère amie Ouendi lynda pour son soutien et sa présence durant tout le parcours.

Enfin, je remercie ma famille qui m'a tant supporté tout au long de ce travail : maman, papa à qui je dois mon succès, je vous dédie cette thèse, mes frères Rabah, Jugurtha et Tarik, ma très chère grand-mère ainsi que mon oncle et sa famille.

Sommaire

Introduction générale 4

Chapitre I: Introduction aux systèmes d'information

1. Introduction	1
2. Notion de base.....	1
2.1 Données.....	1
2.2 Information	2
2.3 Système	2
2.4 Document	3
2.5 Système d'information.....	3
3. Système d'information et système informatique.....	6
4. Fonction d'un système d'information.....	7
5. Composants d'un système d'information	9
6. Typologie des systèmes d'information	9
7. Architecture des systèmes d'information	10
7.1 Exemples d'architecture de système d'information.....	12
7.2 L'architecture orientée services.....	13
7.3 Les éléments de base de l'architecture	13
7.4 Les apports de SOA et l'interopérabilité.....	14
7.5 Scénario d'exploitation de SOA	15
Conclusion.....	16

Chapitre II: Historique des systèmes d'information

1. Introduction.....	17
2. Historique.....	17
2.1 Les différentes phases du système d'information	18
2.1.1 Première phase - la production de masse : le batch.....	18
2.1.1.1 Batch comme choix unique	18
2.1.1.2 L'IBM 360, le méga projet qui structura l'industrie de l'informatique durant vingt ans	18

2.1.1.3	Groupe de travail pour sortir de l’impasse : le SPREAD.....	19
2.1.2	Deuxième phase – « la production en temps réel » : transactionnel et réseau.....	19
2.1.2.1	Passage du batch au transactionnel.....	20
2.1.2.2	Émergence du transactionnel	20
2.1.2.3	Conséquences sur les systèmes d'exploitation des ordinateurs.....	21
2.1.2.4	Affrontement du mode bloc/mode caractère.....	23
2.1.2.5	Crise du logiciel.....	24
2.1.2.6	Echec des L4G	27
2.1.2.7	Bref retour sur ‘l’épisode noir’ de l’IA	28
2.1.3	Troisième phase – l’utilisateur : document et Gui	29
2.1.3.1	Le revamping pour faire « comme si ».....	31
2.1.3.1.1	Le client/serveur	31
2.1.3.1.2	Faire du neuf avec du vieux.....	31
2.1.3.1.3	Aller au-delà du revamping de surface.....	32
2.1.3.1.4	Le mode graphique, la chance du client/serveur.....	33
2.1.3.1.5	Le client/serveur de données domine.....	34
2.1.3.1.6	Le SGBDR s’implémente au cœur du système d’information.....	34
2.1.4	Quatrième phase – donner un sens aux données : le décisionnel.....	35
2.1.4.1	Situation débloquent par le client-serveur	36
2.1.4.2	De l’infocentre au DataWarehouse.....	36
2.1.5	Cinquième phase - l’intranet : informations et connaissances.....	37
2.1.5.1	L’intranet.....	38
2.1.5.2	Le web débloquent la situation	38
2.1.5.3	Émergence et divergence du “Web client-serveur”	38
2.1.5.4	Une alternative au Protocol HTTP	39

2.1.5.5 L'échec du java Computing	39
2.1.5.6 L'intranet consacre l'interface utilisateur du web	40
2.1.6 Sixième phase – l'extranet : optimisation par l'entreprise élargie.....	41
2.1.6.1 Du système d'information aux systèmes de communication.....	41
2.1.6.2 Le traumatisme de l'an 2000	41
2.1.6.3 Défis des systèmes d'information d'aujourd'hui	42
Conclusion.....	43

Chapitre III: Urbanisation des systèmes d'information

1. Introduction.....	44
2. La problématique du management du système d'information	44
2.1 Informatisation des entreprises et systèmes d'information.....	44
2.2 Bilan de l'évolution	45
2.2.1 Un système d'information sédimenté.....	45
2.2.2 Un besoin de cohérence	46
2.2 Les notions d'architectures	47
2.2.1 Analogies et parallélismes.....	47
2.2.1.1 Le projet informatique comme un chantier	47
2.2.1.2 L'architecte coordinateur	47
2.2.2 Définitions	48
2.2.2.1 L'architecture métier.....	49
2.2.2.2 L'architecture fonctionnelle.....	49
2.2.2.3 L'architecture applicative.....	49
2.2.2.4 L'architecture technique.....	50
2.2.3 Cohérence des architectures.....	15
2.3 Le concept d'urbanisation des systèmes d'information.....	51
2.3.1 La métaphore de la ville	51
2.3.2 Définition.....	52
2.3.2.1 Urbanisme de la cité	52
2.3.2.2 Urbanisme des systèmes d'information.....	52
2.3.2.3 Architecture et urbanisme	53
2.3.3 Les limites de l'urbanisme des SI.....	54
2.3.4 Démarches d'urbanisation et règles d'urbanisme	54
2.3.4.1 Présentations des démarches.....	54
2.3.4.2 Le Plan d'Occupation des Sols	55
2.3.4.3 La définition des sous-ensembles.....	55
2.3.4.4 L'élaboration de règles d'urbanisme.....	57
3 Cartographies des systèmes d'information	57

3.1 Modélisation du système d'information	57
3.1.1 Pourquoi modéliser ?	58
3.1.2 Que modéliser dans le cadre de l'urbanisation ?.....	59
3.1.3 Comment modéliser ?	59
3.2 Cartographies et démarches	60
3.2.1 Différents types de cartographies	60
3.2.1.1 Cartographeur l'architecture métier	60
3.2.1.2 Cartographeur l'architecture fonctionnelle.....	61
3.2.1.3 Cartographeur l'architecture applicative	61
3.2.1.4 Cartographeur l'architecture technique	63
3.2.2 Démarches	63
3.2.2.1 Première démarche : une action ponctuelle.....	64
3.2.2.2 Seconde démarche : une action récurrente	64
3.2.2.3 Faire vivre la cartographie	65
3.3 Intérêt des cartographies pour le pilotage de l'évolution du SI.....	65
3.3.1 Positionnement des cartographies dans le cadre de l'urbanisation.....	66
3.3.2 Avantages pour piloter l'évolution du système d'information.....	66
3.4 Contexte de la cartographie.....	67
3.4.1 Situation de départ	67
3.4.1.1 Motivation de la cartographie.....	67
3.4.1.2 Première version de cartographie.....	68
3.4.1.3 Problématique	69
3.5 La cartographie s'inscrit dans le cadre de l'urbanisation.....	70
Conclusion	71
Glossaire	72
Bibliographie	76
Annexes	92
Annexe 01 : Présentation du logiciel MEGA	82
Annexe 02 : Les 7 règles d'urbanisme de Jacques Sassoone.....	83
Annexe 03 : Les diagrammes d'UML.....	88
Annexe 04 : La démarche d'urbanisation de Gérard Jean	89
Annexe 05 : La démarche d'urbanisation de Christophe Longépé.....	91
Conclusion générale	

Table des illustrations

Figure 01 : vue systémique d'un système d'information (Modèle O.I.D).....	5
Figure 02 : structure du système d'information.....	6
Figure 03 : fonction système d'information.....	8
Figure 04 : Décomposition d'une application.....	11
Figure 05 : SOA pour encapsuler la complexité des applications.....	15
Figure 06 : Représentation de scénario d'exploitation de SOA.....	16
Figure 07 : la courbe hype.....	29
Figure 08 : Evolution des systèmes d'information.....	46
Figure 09 : Zones du système d'information dans les banques.....	56
Figure 10 : Exemple de cartographie des processus : le cas d'une SSII.....	61
Figure 11 : Exemple simplifié de cartographie applicative : le cas d'un tour opérateur.....	63
Figure 12 : Matrice du cadre général des travaux d'urbanisation.....	66
Figure 13: Représentation matricielle des applications.....	68
Figure 14 - Illustration de la règle d'appartenance	84
Figure 15 - Illustration de la règle d'autonomie	84
Figure 16 - Illustration de la règle d'asynchronisme	85
Figure 17 - Illustration de la règle des points d'ancrage	86
Figure 18 - Le gestionnaire de flux	87
Figure 19 - Démarche d'urbanisation de Gérard Jean.....	90
Figure 20 - Démarche d'urbanisation de Christophe Longépé.....	91

Introduction générale

Introduction générale :

L'information constitue de nos jours un enjeu considérable d'influence et de pouvoir pour l'entreprise. De par son importance due entre autre à sa nécessité lors de la prise de décision, elle confère aux systèmes d'information une place très privilégiée au sein des entreprises.

Les citations suivantes, extraite d'un ouvrage d'Humbert Lesca et qui sont des témoignages de chefs d'entreprises réussies, illustrent bien cela :

- « L'information et le savoir ne sont pas une fin en soi, mais sans eux comment assurer le professionnalisme nécessaire à l'adaptation que requiert l'ouverture de l'entreprise sur le monde et le futur ? ».
- « L'information est considérée comme la ressource des ressources. Elle est la première condition pour la prise de décision et le moyen d'assurer l'ouverture et l'adaptation permanente de l'entreprise à son environnement ».

"le système d'information est un ensemble organisé de ressources : matériel, logiciel, personnel, données, procédures permettant d'acquérir, traiter, stocker, communiquer des informations (sous forme de données, textes, images, sons etc) dans les organisations" [Reix,04]. Le système d'information est une infrastructure informationnelle supportée par des applications informatiques et utilisée par les acteurs pour leurs besoins de production et de gestion.

C'est ce qui a motivé, notre choix pour le thème –urbanisation des systèmes d'information- comme sujet de recherche et qui représente un domaine d'étude très récent et vaste.

Après avoir introduit la notion de système d'information dans le premier chapitre et cela à travers les différentes notions de base, ces composants ainsi que son architecture, nous nous sommes intéressés de plus près à ces systèmes en retraçant leur histoire et en rapportant les événements les plus marquant, ce qui alors nous a permis de mieux comprendre leur évolution et d'en déduire de ce fait les problèmes majeurs qui empêche leur développement et leur continuité et ainsi pouvoir éviter de retomber dans les même situation de blocage.

Nous proposons ensuite, dans le chapitre trois, la solution aux problèmes rencontrés dans le précédant chapitre et qui a permis aux systèmes d'information d'aller de l'avant. Ce qui nous amène donc à la notion d'urbanisation : une discipline s'intéressant à l'évolution des systèmes d'information et qui ambitionne de doter les organisations de systèmes d'information adaptatifs, pour les mettre plus complètement au service de la stratégie.

Nous concluons avec quelques perspectives qui peuvent être sujettes à de futurs thèmes de recherches et qui y contribueront à améliorer considérablement les systèmes d'information.

Chapitre I

Introduction aux Systèmes d'Information

Chapitre I : Introduction aux systèmes d'information

1. Introduction :

L'*information* n'a jamais eu autant de poids dans notre société qu'aujourd'hui. Dans un environnement de plus en plus complexe, dans un contexte économique particulièrement tendu et en constant mouvement, la performance des entreprises dépend de plus en plus de leur capacité à récupérer la bonne *information* au bon moment et à savoir la partager. Il y a peu, l'*information* était synonyme de pouvoir et la détenir était souvent considéré comme une valeur positive : aujourd'hui elle doit non seulement irriguer l'entreprise mais aussi se propager au sein de celle-ci, à destination des clients, des fournisseurs, des partenaires ou encore des actionnaires. Elle confère aux systèmes d'information une place très privilégiée au sein des entreprises.

Un *système d'information* est "un ensemble organisé de ressources: matérielles, logiciels, personnels, données, procédures permettant d'acquérir, traiter, stocker, communiquer des *informations* (sous forme de données, textes, images, sons, etc....) dans des organisations" [Reix, 2004].

Nous prenons ici le terme " système d'information " de manière assez générale ; en effet, les systèmes d'information, en fonction de leur finalité, appartiennent à diverses catégories : systèmes de production, systèmes de décision, systèmes documentaires, ...; nous prendrons donc le terme dans son acceptation globale. Par ailleurs, il convient de définir quelque peu la " matière " traitée, mémorisée et communiquée par le système d'information. Cette matière est définie sous des vocables divers dont les principaux sont les données, les informations, les connaissances, et les documents.

2. Notions de base :

2.1 Données:

Une donnée est une représentation codée d'un fait dans un but de communication ou de traitement et manipulable par un système informatique.

Les données sont régies par des règles qui peuvent être seulement opératoires ou issues d'une expertise plus ou moins complexe. L'ensemble des données et des règles qui les régissent correspondent à la notion de connaissance. [2]

La donnée est d'après le Robert "une représentation conventionnelle d'une information sous une forme permettant d'en faciliter le traitement automatique" et l'information "Élément de connaissance susceptible d'être représenté à l'aide de conventions pour être conservé, traité ou communiqué ".

L'AFNOR [Cha 93] pour son compte propose les définitions suivantes : "une donnée comme un fait, notion ou instruction représentée sous forme conventionnelle convenant à la communication, à l'interprétation ou au traitement par des moyens humains ou automatiques".

Chapitre I : Introduction aux systèmes d'information

Et "une information est la signification que l'homme attache aux données au moyen de conventions connues utilisées dans sa représentation".

2.2 Information:

En ce qui nous concerne, l'information est tout renseignement qui nous apporte une connaissance, pouvant être perçu par l'un des cinq sens humains, et qui modifie notre vision du monde et qui réduit notre incertitude sur un domaine précis. Une information n'est pas une donnée, elle est plutôt l'interprétation transmise par une ou plusieurs données, tandis qu'une donnée désigne l'ensemble des symboles représentatifs d'une information.

L'information constitue de nos jours une ressource stratégique pour l'entreprise au point où celle-ci n'y survivrait pas sans. [Gui 91] avait considéré l'information comme un processus informant et informé grâce auquel l'entreprise s'informe et informe. Elle s'informe sur elle-même et sur son environnement et réciproquement informe son environnement sur elle-même. On peut aussi dire que la donnée est la matière première tandis que la l'information est le produit fini (ou résultat).

2.3 Système : [Izz, 2000]

Qu'est ce qu'un système ? on peut définir un système comme un ensemble d'éléments qui sont en corrélation ou en interaction et qui forment un tout. On trouve un grand nombre d'exemples de système en physique ou en biologie, dans les techniques modernes et dans la société. Sur le plan physique, il y a le système solaire qui comprend les planètes, le système biologique du corps humain, le système technologique d'une raffinerie de pétrole et le système socio-économique d'une entreprise. Voici une définition générique qui sert à orienter plus clairement l'étude des systèmes d'information :

Un système est un groupe de composantes reliées œuvrant à un objectif commun dans un processus de transformation organisé qui utilise des ressources et les transforme en produits final.

Un tel système (parfois appelé « système dynamique ») possède trois composantes ou fonctions qui sont en interaction :

- § **Les entrées** sont les éléments que l'on rassemble afin de les introduire dans le système où ils subiront un processus de traitement (ex : des matières premières, de l'énergie, des données et un effort humain).
- § **Le traitement** est le processus qui permet de convertir les entrées en résultats (ex : la respiration ou les calculs mathématiques).
- § **Les sorties** sont les éléments résultant du processus de transformations (ex : les produits finis, les services humains et les rapports présentés aux dirigeants).

Cette définition montre deux conditions importantes sans lesquelles quelque chose ne peut être qualifié de système et qui sont respectivement la détermination d'un objet et l'existence d'interactions entre les éléments

Chapitre I : Introduction aux systèmes d'information

2.4 Document :

Dans son acception courante un document est généralement défini comme le support physique d'une information.

Les documents édités sont destinés à une diffusion plus ou moins large. Les plus répandus sont :

- les livres
- les journaux et les périodiques
- les enregistrements sonores (cassette et disques)
- les enregistrements vidéo (cassette et DVD)
- les cartes et plans
- les partitions
- les documents électroniques

D'autres documents sont d'abord destinés à un usage précis :

- les correspondances
- les imprimés, factures, formulaires, billets, menus, étiquettes, etc.
- les notes manuscrites ou dactylographiées
- les photographies

Lorsqu'ils sont conservés au-delà de leur durée d'usage originel, ils deviennent des *documents d'archive* et sont une source pour l'historien.

Selon l'encyclopédie scientifique, en Sciences de l'Information, le mot document correspond plus précisément à la 'réunion' d'un support physique et d'une information (pertinente pour l'utilisateur).

En informatique, le mot document ou document électronique est généralement synonyme de *fichier*, bien que la notion de fichier soit plus large.

2.5 Système d'information :

Définition 1 : [Reix, 2004]

"Un système d'information est un ensemble des ressources : Matériel, logiciel, personnel, données, procédures permettant d'acquérir, de traiter, stocker, communiquer des informations (sous forme de données, texte, image, sons, etc.) dans des organisations".

Cette définition, tout comme celle de Gérard Jean, met en avant le contenu procédural du SI, en insistant sur les interactions entre ses éléments constitutifs. De la même façon, Christophe Longépé dans son ouvrage sur le projet d'urbanisation [*Longépé, 01 (1)*], définit le SI comme "l'ensemble des moyens mis en œuvre pour stocker, traiter, générer et restituer les informations nécessaires au bon fonctionnement de l'entreprise ou de l'organisme". Il insiste là encore sur les interactions des informations.

Chapitre I : Introduction aux systèmes d'information

Définition 2 : [Bouchy, 94]

" Un SI est constitué d'activités, de communications, de dialogues, d'entrées et de sorties, de règles d'organisation, qui sont des objets externes et de données, de traitements, de synchronisations, de ressources, etc, qui sont des objets internes, les uns et les autres étant composant élémentaires qui participent tous à la description du système considéré"

Définition 3 : [Rolland, 1990]

"Un SI est un artefact, un objet artificiel, greffé sur un objet naturel qui peut être une organisation, il est conçu pour mémoriser un ensemble d'images de l'objet réel à différents moments de sa vie, ces images devant être accessible par le partenaire de l'organisation qui s'en servent pour décider des actions à entreprendre dans les meilleures conditions".

Un système d'information est donc, en quelque sorte, une extension de la mémoire humaine qui amplifie le pouvoir de mémorisation des acteurs de l'organisation et facilite leur prise de décision.

Définition 4 : [CASTELLANI, 1987]

"Un SI est constitué :

- Ø D'un ensemble de communication (et des données qu'il communique) ;
- Ø D'un système de traitement (éventuellement) ;
- Ø D'un système de données résidentes (éventuellement)

Un système de communication (d'un système d'information) est un système :

- Ø Qui transmet les données entre l'homme et/ou des machines ;
- Ø Et qui assure éventuellement une mémorisation temporaire de ces données ».

Définition 5: [Darras, 2004]

On peut distinguer trois points de vue complémentaires du SI :

- Ø Un point de vue fonctionnel pour lequel le SI est :
 - § Capable de collecter l'information provenant de la partie opérante,
 - § Capable d'organiser cette information en vue de son traitement,
 - § Capable de publier cette information à destination des utilisateurs concernés (utilisateurs ou autres systèmes),
 - § Capable de mémoriser l'information, mais aussi de la connaissance sur l'organisation du travail,
 - § Capable d'aider les acteurs de l'organisation dans l'exercice de leur métier, en particulier supporter la décision au sein de l'organisation.

Chapitre I: Introduction aux systèmes d'information

- Ø Un point de vue systémique le SI peut être :
 - § Perçu comme le sous-système de l'organisation

Donc un système qui englobe tous les composants dont les interactions sont de type informationnel. Son objectif est de fournir aux différents niveaux de l'organisation les informations nécessaires permettant de la faire tourner [Erm, 1994]. On retrouve dans ce même courant d'idée la définition de J.Lemoigne qui aborde les systèmes d'information avec une approche systémique qui consiste à définir une organisation comme la composition de trois types de systèmes qui sont :

- **Le système opérant** : qui réagit aux événements quotidiens, qui viennent de l'environnement, selon les règles définit. Il est chargé de transformer des ressources ou flux primaires (flux financier, flux de personnel, flux de matière, flux d'information).
- **Le système de pilotage** : qui permet d'engager le processus de décision tout en définissant au préalable les objectifs, les critères d'évaluation et les règles de gestion. Il dirige l'organisation et maintient le cap sur les objectifs choisis.
- **Le système d'information** : qui relie les deux systèmes précédents tout en jouant un rôle de coupleur. Il correspond à la partie chargée de la collecte, du traitement, du stockage et de la diffusion des informations. Il est capable d'une part de contrôler le déroulement de différents processus d'une entreprise et, d'autre part, de fournir aux gestionnaires les informations sur l'état de la structure pilotée et sur l'environnement de l'entreprise.

La figure suivante [Le Moigne, 1999] montre le système d'information couplé au système opérant et au système de pilotage.

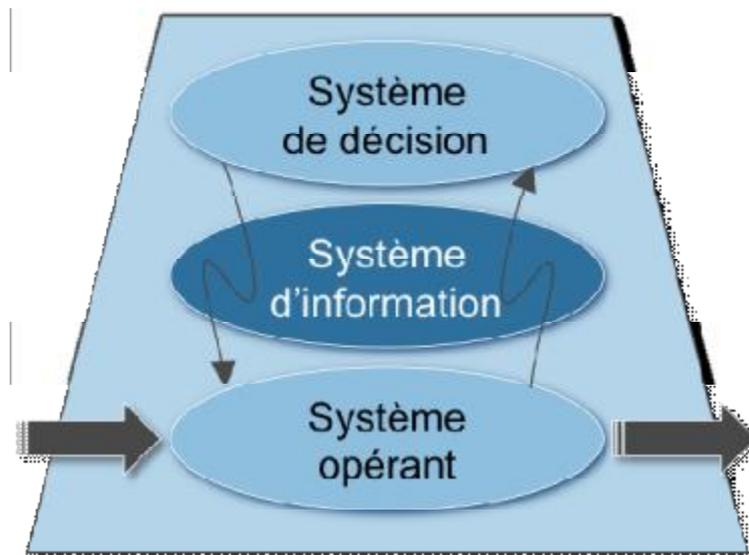


Figure 01 : vue systémique d'un système d'information
(Modèle O.I.D)

Chapitre I : Introduction aux systèmes d'information

- ∅ Un point de vue structurel pour lequel le SI est :
 - § Une suite d'interaction entre l'homme et les moyens informatiques,
 - § Décomposable au niveau des moyens en une collection d'entités matérielles et logicielles dont l'agencement est variable, et de plus en plus désigné par le terme architecture.

Le point de vue structurel décomposé un SI en deux sous-système : le système de traitement de l'information (comprenant les acteurs, les données et les processus) et le système informatique (comprenant les ressources matérielles et logicielles, les bases de données et les fonctions. Cette décomposition éclaire la relation entre les technologies et pratique d'usage de l'information.

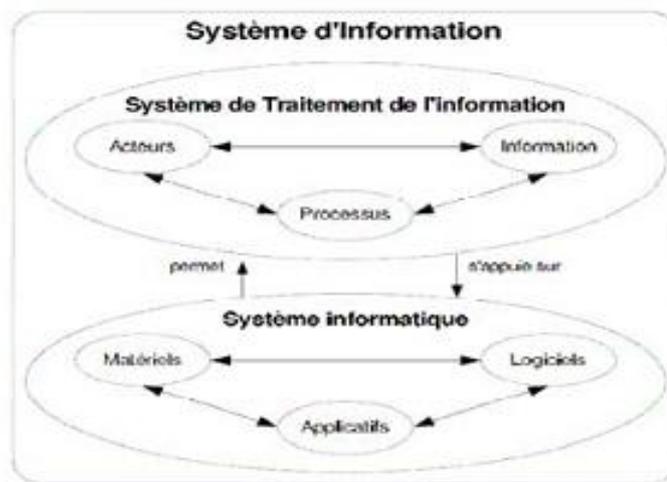


Figure 02 : structure du système d'information [Morley et al, 2005]

En reprenant les points de description d'un système, on a :

- Les éléments constitutifs du système d'information sont les informations*.
- Ces informations interagissent par le biais des processus et des procédures de l'entreprise.
- L'objectif du système d'information est de structurer les informations pour servir le métier et ses acteurs.

3. Système d'information et système informatique :

Le système information n'est pas assimilable au système informatique qui n'en est qu'une partie.

Le système d'information est essentiellement sémantique et fonctionnel ; il est défini par le " modèle complet " en langage UML, qui fait abstraction des moyens techniques. La maîtrise

Chapitre I : Introduction aux systèmes d'information

d'ouvrage doit se doter d'une expertise spécifique, fonctionnelle, garantissant la pertinence des demandes en regard des exigences du métier.

Le terme d'informatique est un néologisme construit à partir des mots information et automatique par Philippe Dreyfus¹ en 1962. Il s'agit donc d'une discipline qui concerne le traitement automatique de l'**Erreur ! Source du renvoi introuvable.***. La définition acceptée par l'Académie Française est la suivante :

"Science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications dans les domaines techniques, économiques et sociaux".

La définition proposée par Gérard Jean pour le système informatique est la suivante :

"ensemble structuré des applications, des données et des infrastructures automatisant, pour tout ou partie, les fonctions et informations" [Jean 00].

Cette définition répond à l'ensemble des aspects présentés dans la définition du système :

- Les éléments constitutifs du système informatique sont les applications, les données et les infrastructures associées.
- Les interactions entre ces applications sont les échanges de données entre applications, rendus possible par les réseaux et les interfaces.
- L'objectif du système informatique est l'automatisation du traitement et de la gestion des informations.

Le système informatique donc organise ces moyens ; il choisit les plates-formes, langages, interfaces, architectures (centralisée, client serveur à deux ou trois niveaux), la localisation des traitements et mémoires, les niveaux de conservation des données, la couche de middleware et la gestion de la persistance. Il repose sur une expertise attentive à la diversité des outils du marché, aux innovations, à la pérennité des solutions.

4. Fonction d'un système d'information :

Un système d'information collecte, diffuse, transforme et stocke des données pour fournir les informations nécessaires à un acteur ou un groupe d'acteurs [Vidal, 2009].

Cette définition identifie les quatre fonctions endogènes du système d'information.

¹ Philippe Dreyfus fut en 1951 directeur du centre de calcul de Bull. Le mot "informatique" est né d'un jeu de mots qu'il cherchait un nom pour une nouvelle société dont il était l'initiateur. Cette société fut nommée SIA (Société d'Informatique Appliquée).

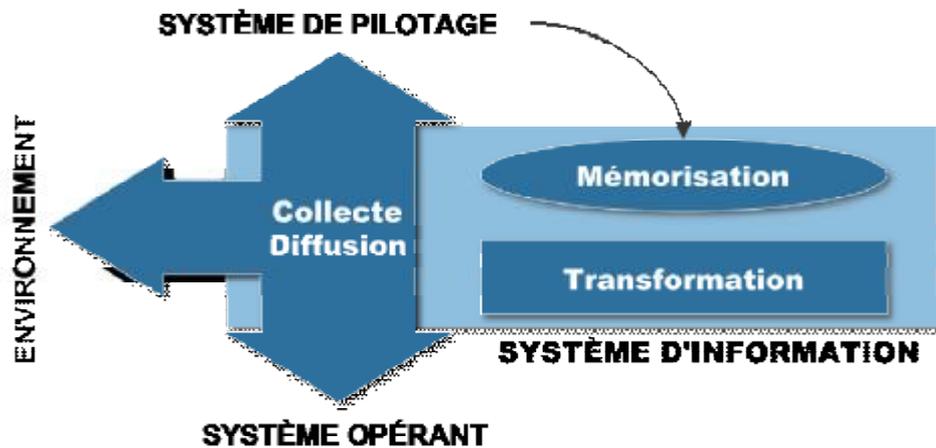


Figure 03 : fonction système d'information [Vidal, 2009]

La collecte : est la fonction par laquelle le système d'information acquiert les informations provenant de l'environnement, du système de pilotage ou du système opérant.

Le stockage : est la fonction par laquelle le système d'information conserve les informations qu'il manipule. Les informations se présentent sous forme de données et la mémorisation de ces données est réalisée aussi bien par des outils manuels que par des outils informatiques. L'évolution technologique a beaucoup influencé cette fonction du fait d'une part de l'augmentation des capacités de stockage et, d'autre part, de la diminution des coûts de stockage. Par exemple, les bases de données et les entrepôts de données sont devenues des supports « incontournables » aux différentes applications qui supportent le système d'information d'une organisation.

La transformation : est la fonction qui permet au système d'information de construire de nouvelles informations en modifiant leur fond ou leur forme.

Cette fonction s'appuie sur des opérations telles que la comparaison, le calcul et la transformation de format. L'ensemble des traitements constitue ce que certaines méthodes comme MERISE appellent le processeur d'informations [Mat, 1995]. Là encore, l'évolution technologique a fortement influencé cette fonction en automatisant certains traitements.

La diffusion : est la fonction par laquelle le système d'information transmet les informations vers l'environnement, le système de pilotage ou le système opérant.

Cette fonction s'appuie sur les mêmes avancées technologiques que la fonction de collecte.

L'informatique joue un très grand rôle dans la réalisation de ces quatre fonctions.

Chapitre I : Introduction aux systèmes d'information

5. Composants d'un système d'information : [O'Bri, 1997]

Un système d'information utilise des ressources humaines (utilisateurs finals et informaticiens), du matériel (machines et supports) et des logiciels (programmes et procédures) pour accomplir des fonctions de saisie, de traitement, de stockage et de contrôle qui servent à convertir en produits informatifs des ressources en données.

§ **Les ressources humaines** : le fonctionnement des SI exige le concours des ressources humaines. Ces ressources humaines se divisent en deux catégories : les utilisateurs finals et les informaticiens.

-*les utilisateurs* : sont des personnes qui utilisent un SI ou l'information qu'il produit. Il peut s'agir de comptables, de représentants, d'ingénieurs, de clients ou de dirigeants.

-*les informaticiens* : sont des personnes qui conçoivent des SI, et qui les exploitent. Les informaticiens regroupent les analystes fonctionnels (conçoivent des SI en fonction des besoins des utilisateurs), les programmeurs (élaborent des programmes informatiques à partir des exigences des analystes), les opérateurs sur ordinateurs (exploitent les SI) et certains cadres, techniciens et employés de bureau.

§ **Les ressources matérielles** : englobent tout les dispositifs physiques et toutes les machines qui servent au traitement de l'information. Cette notion comprend non seulement les machines, comme les ordinateurs et les calculateurs, mais aussi les supports, c'est-à-dire les objets sur lesquels on enregistre les données comme les disques magnétiques.

§ **Les ressources logicielles** : recouvre l'ensemble des instructions de traitement de l'information. La notion de logiciel comprend non seulement les instructions d'exploitation que nous appelons *programmes*, lesquels gèrent le matériel informatique et le contrôlent, mais aussi les ensembles d'instructions de traitement de l'information que nous appelons *procédures*.

§ **Les ressources en données** : représente bien plus que la simple matière première des SI. C'est pour quoi les gestionnaires et les informaticiens ont élargi la notion de ressources en données : ils constatent qu'elles constituent aussi une ressource organisationnelle précieuse.

6. Typologie des systèmes d'information : [O'Bri, 1997]

Il existe plusieurs classifications des SI, cela d'après les penseurs et auteurs. Mais tous spéculent autour de quatre grands types de SI :

- Le système de traitement des transactions (STT)
- Le système de reportage de l'information (SRI)
- Le système d'aide à la décision (SAD)
- Le système de décision programmée (SDP)

Analysons ces systèmes en détail.

§ **Système de traitement de transaction (STT)**

Chapitre I : Introduction aux systèmes d'information

Un STT collecte, mémorise, et traite des données décrivant les transactions d'une organisation de façon que les membres de l'organisation en question puissent s'en servir comme base de leurs actions régulières, de type : procédure, production des documents administratifs, utilisation dans d'autres systèmes, ...

Ce système est aussi généralement celui qui supporte la cueillette, le traitement et la mémorisation des données externes à l'organisation. Ex. : en vente, un système de traitement de commandes, collecte des données sur les clients, mémorise ces données, les traite, soit pour analyser les modalités de distribution des produits, soit pour préparer une information qui sera utilisée plus tard par d'autres systèmes par une prise de décisions.

§ Système de reportage de l'information (SRI)

Ce système a pour finalité de produire des rapports prédéfinis pour les membres d'une organisation à partir des traitements des données déjà cueillies et mémorisées. Notons qu'il n'existe pas des contraintes sur la nature de ces rapports, sur leurs fréquences ou encore sur les événements qui causent leur production.

Ces rapports peuvent être imprimés sur papier, affichés sur un écran cathodique, etc. ils peuvent être produits quotidiennement, hebdomadairement, régulièrement ou sur commande. Ex. : en gestion des ressources humaines, un rapport sur l'arrivée et le départ du personnel au site de travail.

§ Systèmes d'aide à la décision (SAD)

De l'anglais « decision support system », il supporte activement le processus décisionnel de l'entreprise.

C'est la suite logique des systèmes de production de rapport et de traitement transactionnel. Ce sont des SI interactifs et informatisés qui font appel à des modèles décisionnels et aux bases de données pertinentes en vue d'aider les gestionnaires à prendre des décisions. Ils permettent la modélisation analytique, l'extraction de données et la représentation graphique de l'information. Les dirigeants s'en servent lorsqu'ils ont besoin d'information pour prendre des décisions non-structurées dans un cadre interactif et informatisé.

§ Système de décision programme (SDP)

Ce système remplace l'humain (décideur) dans le processus décisionnel. Ainsi, on fixe des règles intégrées au logiciel qui, dans certaines conditions, provoquent les actions du système. Ceci s'oppose au STT et au SAD utilisant l'homme comme décideur.

Le rôle d'un SI est de produire de l'information et/ou d'assister ou d'automatiser le travail par d'autre système de travail. Un SI peut servir d'autres systèmes de travail de différentes manières.

7. Architecture des systèmes d'information : [Biga, 2006]

Une application informatique est constituée de trois couches principales : l'*interface* avec les utilisateurs, les *traitements* et les *données* (voir figure ci-dessous)

Chapitre I : Introduction aux systèmes d'information

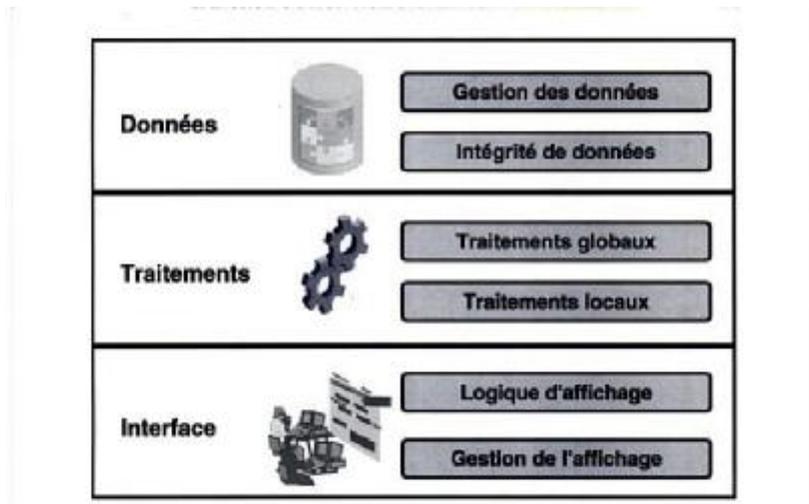


Figure 04 : Décomposition d'une application

Chaque couche peut se décomposer en deux modules distincts. Ainsi, six modules se dégagent d'une application informatique.

La couche données est constituée de modules suivants :

- la gestion des données* : elle assure la sélection, l'accès aux données ;
- l'intégrité des données* : elle concerne tout ce qui est lié à la sécurité des données (partage et accès, valeurs,...).

La couche traitement contient la logique applicative c'est-à-dire les traitements à réaliser par l'application. Ils peuvent être découpés en deux familles :

Les : *traitements locaux* et *traitements globaux*

-les premiers, regroupant les contrôles effectués au niveau du dialogue avec l'IHM visant essentiellement le contrôle et l'aide à la saisie ;

-les seconds constituant l'application elle-même. Cette couche, appelée couche métier, contient les règles internes qui régissent une entreprise donnée.

La couche interface est composée des modules suivants :

- la logique de l'affichage* : elle transmet à la gestion de l'affichage la description des éléments de présentation ;
- la gestion de l'affichage* : elle est assurée par l'environnement de l'utilisateur.

On peut notamment dire que L'architecture des systèmes d'information est une discipline multiforme et transversale qui traite de la démarche, des modèles, des outils et des langages permettant de concevoir et construire un système d'information. Elle vise à décrire la structuration d'un système d'information en termes de constituants et d'organisation de ces constituants en tenant compte des multiples niveaux de conception, des aspects de

Chapitre I : Introduction aux systèmes d'information

modélisation, mais aussi du degré de généralité du système. Dans [Morley et al. 05], on identifie une architecture de système d'information comme « une représentation abstraite des différentes parties du système d'information qui permet des décisions globales et de s'assurer de la pertinence de l'assemblage, notamment la cohérence et l'efficacité technique... ».

L'architecture représente alors un enjeu fondamental en vue de permettre une certaine maîtrise de la conception et de l'évolution des systèmes d'information. La maîtrise de la conception passe par une description détaillée des composants, niveaux et organisations mais également par différents modèles de conception d'un système d'information. La maîtrise de l'évolution, comme la définit [Vernadat 06] passe par une gestion efficace des changements qui interviennent durant le cycle de vie du système.

Nous ne pouvons pas parler d'architecture de système d'information sans évoquer l'architecture de l'entreprise. Nous dirons synthétiquement que l'architecture de l'entreprise et l'architecture de son système d'information sont étroitement liées. En effet, pour évoluer sur un marché fortement concurrentiel, les entreprises doivent s'adapter aux évolutions du contexte technique et du contexte métier simultanément. Ceci passe nécessairement par la définition d'une architecture flexible de leurs systèmes.

7.1 Exemples d'architecture de système d'information

Il existe une typologie riche d'architectures de système d'information. Ces architectures présentent des niveaux d'abstraction différents d'un système d'information. Nous distinguons:

- **architecture fonctionnelle** : permet de spécifier les besoins fonctionnels des utilisateurs de système d'information. Ainsi, un système d'information est présenté en termes de fonctions, liées entre elles par des flux d'entrée/sortie (SADT) ou liées aux acteurs qui les manipulent (diagramme de cas d'utilisation UML). Nous parlons aussi d'urbanisme de système d'information, qui consiste à découper le SI en modules autonomes faisant référence aux fonctions de SI. Cette architecture intervient directement pour faire évoluer ou refondre un système d'information.
- **Architecture logique** : permet d'identifier la structuration d'un système d'information en adoptant une logique indépendante des considérations techniques.

Les architectures logiques les plus connues sont : l'architecture orientée composants et l'architecture orientée services (SOA). Les composants ou les services sont caractérisés par le fait qu'ils sont des entités autonomes, réagissent par échange de messages et sont définis sur différents niveaux de granularité. La différence principale entre un service et un composant réside dans le fait qu'une architecture orientée services se concentre sur les exigences déterminées au niveau de la stratégie et du processus métier, alors qu'une architecture orientée composants se concentre sur les composants d'un logiciel. Ces composants peuvent être utilisés pour livrer des services. Un service expose dans ce cas un composant.

- **Architecture informatique (ou physique)** : décrit la structuration d'un système informatique en termes d'organisation de fonctions et des constituants qui le composent. Cette architecture définit d'autres « sous-architectures » :

Chapitre I : Introduction aux systèmes d'information

- Ø **architecture logicielle** : l'agencement et l'interaction des composants logiciels. Ces derniers peuvent être définis sur plusieurs couches (couche présentation, couche applicative, couche données).
- Ø **Architecture matérielle** : l'agencement et l'interaction des composants physiques (disque dur, unité centrale, etc.).
- Ø **Architecture d'intergiciel (Middleware)** : l'agencement et l'interaction des composants servant à faire communiquer plusieurs applications entre elles.
- Ø **Architecture réseau** : l'architecture permettant la communication au sein d'un système d'information.

Synthèse

Les architectures de système d'information ont un rôle capital pour la définition d'un système d'information et pour maintenir son évolution. Il existe donc naturellement un lien fort entre l'architecture de système d'information (conception de SI) et la capacité d'interopérabilité que ce dernier peut avoir.

S'il est tout à fait légitime de présenter MDA dans cette section de chapitre consacrée à l'étude des architectures de système d'information, nous la présenterons dans l'axe « conception dirigée par les modèles ». MDA a permis, en effet, de valoriser l'ingénierie dirigée par les modèles en offrant une architecture structurante. MDA permet de séparer les exigences métiers des spécifications techniques dans une démarche de conception de système d'information.

7.2 L'architecture orientée services

SOA est une architecture logique de système d'information basée sur la notion de « service », qui constitue la brique de base de cette architecture. D'après [Raymond 07] « SOA est un style d'architecture organisé à partir de services métiers communs mutualisés pour un ensemble de lignes métiers ou d'applications ». Un système d'information qui respecte l'architecture SOA est défini en termes de services proposés. SOA peut être utilisée dans un cadre interne à l'entreprise, ainsi les services peuvent être classés par domaine fonctionnel (achat, finance, marketing, etc.), mais surtout SOA présente un moyen efficace pour faciliter la communication avec d'autres entreprises. Dans ce cas, l'entreprise expose (publie) ses services à ses partenaires. Nous pouvons donner l'exemple d'une agence de voyages qui expose ses services de réservation d'hôtel, de vol, de séjour, etc.

7.3 Les éléments de base de l'architecture

F. Vernadat [Vernadat 06] identifie trois concepts de base pour une structuration d'un modèle d'entreprise pour SOA : événement, service et processus. Selon le même auteur, ces concepts doivent être compris et maîtrisés aux niveaux métier (organisationnel) et applicatif (technique). Au niveau métier, ces concepts peuvent être désignés événement métier, service métier et processus métier. Nous présentons ces concepts tels qu'ils sont définis par [Vernadat 06] :

Chapitre I : Introduction aux systèmes d'information

- un évènement métier : est un fait qui se produit en relation avec les opérations de l'entreprise. C'est un changement dans l'état de l'entreprise, qui doit avoir une réponse. Il existe plusieurs types d'évènements : un évènement sollicité (réception d'une nouvelle requête, etc.), un évènement d'exception (panne de machine, etc.), évènement programmé (délai de temps passé, etc.) ou un évènement de synchronisation (début ou fin d'une activité).
- Un processus métier : un processus métier est un séquence ment partiellement ordonné d'un ensemble d'activités et/ou de services dans le but de réaliser un objectif de l'entreprise.
- Un service métier : est une fonctionnalité de l'entreprise qui apparaît atomique du point de vue de l'appelleur de service. Un service doit être identifié d'une manière unique dans l'entreprise.

Cette vision de SOA montre le lien fort entre les services et les processus métier d'une entreprise qui a adopté cette architecture. Dans ce cas, un processus métier peut faire appel à un ensemble de services pour s'exécuter. L'orchestration de services (dite aussi composition) réfère à un processus métier qui est composé uniquement de services.

Toutefois, un service peut exister d'une manière indépendante (cela ressemble à un processus métier composé d'une seule activité). A ces concepts de base, présentés par [Vernadat 06], nous voulons ajouter le concept de message, en effet les services communiquent entre eux en utilisant des messages. Un message présente une structure bien définie. Il est important de respecter cette structure pour pouvoir invoquer un service. Les messages illustrent le point de vue informationnel de SOA.

De nos jours, les Services-web2 fournissent les technologies les plus adaptées pour rendre possible la création de l'architecture orientée services. Cependant, il ne faut pas confondre les Services-web avec SOA. Les Services-web fournissent le support pour la description et l'infrastructure de communication des services, alors que SOA décrit comment un système composé de services peut fonctionner.

7.3 Les apports de SOA et l'interopérabilité

SOA contribue à l'évolutivité, la flexibilité et la pérennité du SI en :

- encapsulant la complexité de ses applications : il s'agit de substituer la découpe strictement applicative d'un SI par une structuration en services plus réduits et potentiellement plus simple à faire évoluer [Raymond 07] (Fig 05).

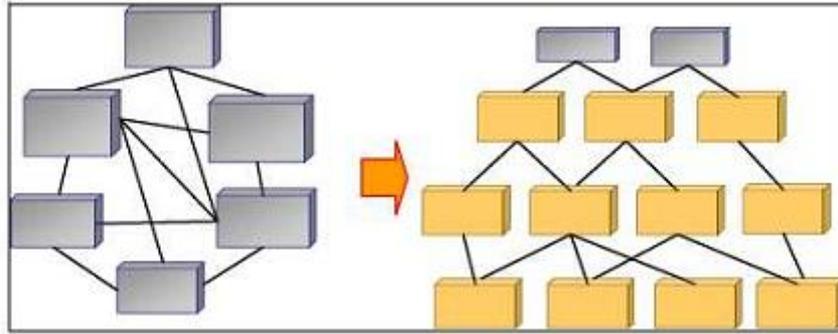


Figure 05 : SOA pour encapsuler la complexité des applications [Raymond 07]

- favorisant son agilité : SOA permet de structurer d'une manière dynamique un système d'information. Ce dynamisme est lié à la possibilité de changer facilement une configuration de services : principe de couplage faible (ajouter ou enlever un service, modifier l'ordre de communication de services). De plus, il est possible d'intervenir sur le contenu de service sans toucher à son interface d'accès (changement interne).
- améliorant son accessibilité : le système d'information est désormais plus facile d'accès, une fois l'exposition des services réalisée. On parle aussi de publication de services. SOA simplifie les échanges inter-entreprises [Monfort et al. 04].

Les trois apports cités ci-dessus (réduction de la complexité, agilité et accessibilité) sont de nature à réduire la difficulté de communiquer entre systèmes d'information. Une difficulté réduite veut dire un moindre effort pour la communication. Rappelons les définitions de l'interopérabilité de [Vernadat 96] : « c'est la capacité de communiquer avec d'autres systèmes et d'accéder à leurs fonctionnalités » et de [Konstantas et al. 05] « c'est la capacité de systèmes à collaborer efficacement sans effort particulier de la part des utilisateurs ». SOA est une réponse conceptuelle adaptée à la problématique de l'interopérabilité.

7.4 Scénario d'exploitation de SOA

La figure (Fig II.5) explique le scénario d'exploitation de SOA. Deux acteurs interviennent dans ce scénario : l'agent fournisseur qui publie les services et l'agent demandeur (ou consommateur) de services.

Les services sont décrits à travers une représentation structurée : XML (eXtensible MarkupLanguage), par l'agent fournisseur. Ensuite, ce dernier enregistre les descriptions de ses services dans un annuaire. L'agent demandeur cherche dans l'annuaire des services selon des critères spécifiques au mécanisme de recherche. L'annuaire retourne à l'agent demandeur

Chapitre I : Introduction aux systèmes d'information

les informations d'un service demandé. L'agent demandeur récupère la description de ce service et l'utilise pour échanger des messages et communiquer avec le service.

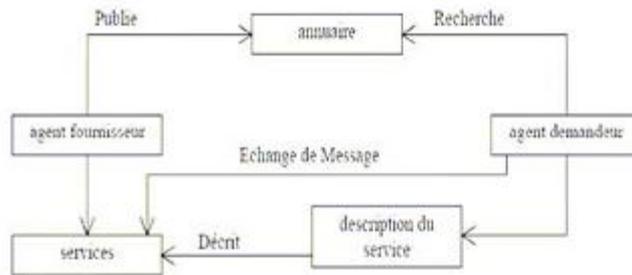


Figure 06 : Représentation de scénario d'exploitation de SOA

Conclusion :

Nous avons dans ce chapitre, pu effectuer un bref panorama sur la notion des systèmes d'information.

Après avoir évoqué les premières notions de bases, nous nous sommes intéressés aux fonctions ainsi qu'aux différents composants qui constitue ces systèmes.

Par la suite on est passé à la typologie et enfin finir avec l'architecture en prenant l'exemple d'une architecture logique qui est l'orienté service.

Dans le chapitre suivant, nous essayerons de retracer le parcours des systèmes d'information et ainsi citer les différentes mutations de ces derniers à travers le temps.

Chapitre I : Introduction aux systèmes d'information

Chapitre II

Historique des Systèmes d'Information

Chapitre II : Historique des Systèmes d'information

1. Introduction :

En dépit de notre obsession pour la nouveauté technologique, les vraies révolutions sont en réalité beaucoup plus rares que l'on s'imagine. Le progrès de l'informatique ne passe pas par des ruptures. C'est plutôt un travail cyclique, où l'industrie des utilisateurs remâche le même concept deux, trois, quatre fois de suite, avant de sortir enfin une version suffisamment robuste pour s'imposer comme standard de fait. [Ief, 2001]

« Ceux qui ignorent les enseignements de l'histoire sont condamnés à le répéter »

C'est tout l'intérêt qu'on peut avoir à analyser le passé pour mieux comprendre le présent et nous aider à tracer ce à quoi ressemble le futur proche.

C'est ce qu'on nous allons essayer d'étudier dans ce présent chapitre qui consiste à retracer le parcours des systèmes d'information à travers le temps.

2. Historique :

Nous pouvons considérer que le système d'information est apparu en même temps que l'écriture. Il existe de nombreux exemples de système d'information dans l'histoire.

En effet, ne peut-on pas considérer que : [1]

- § Les dessins trouvés dans les grottes de Lascaux ont été un moyen d'informer les membres d'une communauté ?
- § Les hiéroglyphes ne constituent-ils un exemple d'écriture utilisée sous l'empire des pharaons ?
- § Le système comptable sumérien n'est-il premier exemple de système comptable ?

Nous pourrions compléter cette liste par de nombreux autres exemples, mais considérons, comme de nombreux auteurs que l'apparition du concept de système d'information est liée à l'introduction des ordinateurs dans les entreprises : année « 0 » des systèmes d'information modernes.

C'est donc en 1954 que le premier ordinateur, utilisé à des fins de gestion, a été introduit dans une entreprise, alors que les premiers développements de l'informatique ont été réalisés à des fins militaires, les dirigeants des entreprises se sont rapidement rendu compte que les applications informatiques pouvaient améliorer le fonctionnement de tout ou partie de l'entreprise.

Pour ce qui est de l'usage systématique de ces ordinateurs dans les organisations c'est seulement à partir de 1960 qu'ils se sont développés. La recherche académique dans le domaine des systèmes d'information émerge à partir de 1965 et s'organise à partir de 1977, c'est donc, à l'échelle de l'histoire des sciences, une discipline jeune [Rodh, 2010].

Chapitre II : Historique des Systèmes d'information

2.1 Les différentes phases du système d'information :

D'après lefebvre [lef, 2001], on peut identifier six mutations qui ont affectés le système d'information. Voici ces différentes phases

2.1.1 Première phase - la production de masse : le batch

2.1.1.1 Batch comme choix unique :

Du point de vue client/utilisateur, le contexte technique de la première époque de l'information moderne est des plus rustiques. La pratique du traitement par lot (batch) n'était pas un choix, c'était la seule possibilité dans la mesure où les systèmes étaient tellement rudimentaire qu'ils n'étaient pas capable de traiter plusieurs tâches en parallèle.

Les organes d'entrées-sorties étaient tous aussi limités, des cartes perforées en entrée et de listings débités par des imprimantes (immenses et bruyantes) en sortie. Au vu des conditions disponibles, les programmeurs de l'époque étaient :

- obligés de se limiter à des chaînes de traitements simplifiées
- les processus pouvaient être découpés en module de calculs successifs.

De plus le développement dans ce contexte était loin d'être facile : non seulement il fallait composer avec des systèmes aux ressources limitées mais la mise au point de nouveaux programmes se faisait presque en aveugle avec des « jeux d'essais » quand il restait du temps machine disponible.

Cette période c'est principalement caractérisé par l'ère du matériel avec le mainframe :

2.1.1.2 L'IBM 360, le méga projet qui structura l'industrie de l'informatique durant vingt ans :

Au début des années soixante, IBM est dans une situation paradoxale.

Son activité informatique commence à vraiment prendre de l'ampleur mais la situation est chaotique à cause de la politique commerciale qui consiste à adresser chaque marché séparément et à les traiter comme des niches distinctes.

De plus, avant 1960, les connaissances des besoins, les modes d'utilisation, des technologies et de leur évolution future n'étaient pas suffisantes pour définir des standards qui auraient pu servir de base à une famille d'ordinateurs universels compatibles.

On constate que le problème était bien présent au niveau matériel mais il était encore plus aigu au niveau logiciel :

- la prolifération des lignes d'ordinateurs avaient engendré la multiplication des combinaisons logicielles associées.

Chapitre II : Historique des Systèmes d'information

- les programmeurs étaient surchargés par la nécessité d'écrire et de réécrire les systèmes d'exploitation pour les différentes familles de systèmes.

Du côté client aussi le problème résidait :

En effet, quand une société avait besoin d'accroître son système informatique au-delà de cette limite, il n'y avait pas d'autres choix que de changer le type de machine. Mais ce passage d'une gamme à une autre impliquait la réécriture des applications précédemment développées.

2.1.1.3 Groupe de travail pour sortir de l'impasse : le SPREAD [Poul, 2011]

En octobre 1961, la direction d'IBM avait nommé un groupe de travail pour le projet de systèmes compatibles.

Ce projet est lancé au début de 1962 et mené sur plusieurs sites dans le plus grand secret.

Fin 63, le développement était en plein boom et la direction commença à réfléchir à la question du lancement. C'est un événement extérieur qui va précipiter le lancement du 360 : L'annonce du modèle H200d'Honeywell.

La gamme fut alors annoncée le 7 avril 1964. [Poul, 2011].

Cette première phase a duré une bonne quinzaine d'années, depuis les années soixante jusqu'au milieu des années soixante-dix.

Durant cette période, les entreprises ont profité de cette informatique pour automatiser la plupart de leurs processus de production ainsi qu'une bonne partie des processus administratifs (comptabilité, paie,...).

Même si c'est dur à admettre, l'essentiel de « l'informatique de production » a été fait à cette époque-là, la suite n'est que raffinement et extension des développements existants.

2.1.2 Deuxième phase – « la production en temps réel » : transactionnel et réseau

C'est une autre période qui a été marquée par les événements suivants :

2.1.2.1 Passage du batch au transactionnel :

C'est caractérisé par le passage d'une informatique très passive (traitement par batch) à une informatique un peu plus dynamique (le transactionnel).

L'avantage est que l'information était disponible plus rapidement et sous une forme plus pratique. Par exemple au lieu de fouiller dans un listing, il suffisait plus tôt d'interroger une base de données à partir de son terminal.

Chapitre II : Historique des Systèmes d'information

Le concept de transaction fait partie du patrimoine universel des sociétés humaines dont il est l'un des éléments régulateurs de la fonction d'échange : échange d'un travail contre un salaire, échange d'un bien ou d'un service contre un autre bien ou service, association de deux individus et/ou organisations en vue d'un but inaccessible à chacune des parties prises individuellement.

Une transaction, selon l'encyclopédie scientifique, implique toujours trois acteurs :

- clients et fournisseurs sont les acteurs de l'échange. Ils procèdent selon un protocole explicite connu des deux parties ; ce protocole peut varier selon la nature de l'échange. Le « client » demande un service ou un bien, le « fournisseur » offre le service ou le bien souhaité ;
- le « juge arbitre », ou observateur neutre, est le garant que la transaction s'est effectuée selon les règles. Il contresigne l'accord des deux parties afin d'en garder une trace officielle accessible par tous.

Il n'est donc pas étonnant de retrouver ce concept au cœur des systèmes d'information dans la mesure où ceux-ci ont comme ambition de modéliser l'activité des individus et/ou des organisations dans le but d'automatiser tout ou partie de la gestion des informations conformément aux objectifs de l'entreprise.

La notion de transaction est donc étroitement associée à celle de déontologie et de respect de règles comportementales. Une transaction est un ensemble d'actions cohérentes qui doivent avoir le même sens pour tous les acteurs : c'est donc une notion fondamentalement sémantique.

2.1.2.2 Émergence du transactionnel

La notion de programmation transactionnelle apparaît explicitement à la fin des années 60, début des années 70, avec les premiers réseaux de terminaux et les premières bases de données. On passe progressivement d'un monde exclusivement batch à celui du télétraitement interactif. Ce faisant, un certain nombre de problèmes systèmes masqués aux programmeurs dans le mode batch deviennent visibles et incontournables pour les programmeurs d'applications interactives.

Dans une chaîne batch, un seul programme est activé par le moniteur de travaux de la machine. En cas d'incidents, il est donc très facile d'annuler le travail en cours et de relancer le programme sur les données initiales. Cela n'a aucun impact sur les usagers du système informatique qui ne sont pas connectés au système.

Dans une chaîne transactionnelle, la situation est totalement différente. Le même programme, la même transaction peuvent être activés par des dizaines, voire des centaines de terminaux dont chacun devra avoir un contexte d'exécution bien à lui, sans possibilité de contamination de ses voisins. En cas d'incidents, il est bien sûr hors de question d'arrêter tout le système car

Chapitre II : Historique des Systèmes d'information

l'attente aux terminaux deviendrait vite insupportable aux usagers du système. La programmation transactionnelle doit gérer beaucoup plus finement les ressources du système informatique : c'est donc une programmation beaucoup plus complexe, tout en étant beaucoup plus fiable vis-à-vis des incidents qui peuvent survenir comme les conflits d'accès aux ressources communes.

En fait, la mise en place dans les entreprises d'un système transactionnel s'effectua en plusieurs phases. La première étant souvent l'utilisation de terminaux pour la saisie interactive d'informations traitées en batch la nuit. Cette saisie permettait de faire l'économie de la phase vérification d'un traitement mécanographique traditionnel. Dans d'autres cas, la priorité était donnée à l'instauration d'un système de consultation (sans mise à jour) d'informations permettant l'accès à certaines bases de données de l'entreprise mises à jour sur une base quotidienne, économisant ainsi l'impression et la diffusion de gros volumes de papier de valeur fugitive. De tels systèmes survivront encore longtemps, le besoin de lire étant supérieur à celui de créer. [3]

La légende historique de l'avantage compétitif fourni par l'informatique transactionnelle est attachée à la compagnie Américaine Airlines avec le fameux système SABRE de réservation de places, qui fut et qui, reste encore l'archétype des systèmes transactionnels..

C'est le développement et la mise au point de SABRE qui est à l'origine de la plus part des progrès techniques qui ont ensuite permis la généralisation du transactionnel et le passage du batch aux traitements partagés en réseau

2.1.2.3 Conséquences sur les systèmes d'exploitation des ordinateurs [3]

Protection

Une des caractéristiques des systèmes transactionnels est que le déploiement des applications est un évènement aussi notable que la mise en service d'une nouvelle usine ou d'une ligne de communication. Le développement et les tests de l'application sont des phases séparées et la maintenance ne peut se faire par des essais de fonctionnement sur le système en service. Il y a donc une différence assez grande entre systèmes transactionnels et systèmes de développements en time-sharing, même si les matériels sont similaires. Il n'est pas question de donner un accès Internet Telnet à un système bancaire, ce qui a comme conséquence d'isoler le réseau applicatif de la plupart des pénétrations externes.

Cependant, il faut qu'un système transactionnel ne s'effondre pas s'il est sollicité par un message saugrenu, erroné volontairement ou non. Il faut donc que l'exécution d'une transaction ne puisse monopoliser des ressources critiques soit directement, soit indirectement par un usage abusif de ressources système partagées.

Multithreading

Le temps de traitement d'une transaction ne dépend pas que du temps pris par l'ordinateur et la transmission des données. Il dépend aussi du temps de dialogue entre l'opérateur et son client à l'occasion des interactions. En effet, si certaines transactions ne comportent qu'une question et une réponse, ce n'est pas le cas le plus général et des choix sont souvent soumis au client en

Chapitre II : Historique des Systèmes d'information

fonction de la disponibilité des stocks d'objets concernés. Il existe même des super-transactions qui comportent des processus asynchrones comme la déconnexion du client et sa reconnexion depuis un autre terminal. Ces super-transactions restaient actuellement gérées selon des protocoles spécifiques à l'application si bien que le rôle du système d'exploitation restait limité à la gestion d'une transaction multi-échanges depuis un même terminal ou du moins depuis la même adresse. Il est évident qu'il est nécessaire de ne monopoliser que le minimum de ressources pour des transactions en attente de la réflexion du client ou celles qui étaient en cours de transmission avant la généralisation des affichages rapides. Mais le temps d'accès aux fichiers sur mémoire de masse représentait tout de même une part importante du temps-ordinateur et très tôt, les concepteurs ont éprouvé le besoin de traiter en parallèle plusieurs transactions.

Une première approche -sur IMS- a été de paralléliser les divers types de transactions en leur affectant à chaque type un serveur et de multiplexer ces serveurs à la manière du multitraitement en batch. Cette approche convient à certaines applications très variées, mais est peu efficace dans le cas des transactions très variées et pour une utilisation en multiprocesseurs. Le programme du serveur doit être réutilisable, mais non nécessairement réentrant.

Une autre approche a été développée à l'origine pour le système SABRE et les successeurs spécialisés APARS de IBM, puis utilisé par CICS et les différents systèmes de Bull et Honeywell (TDS, DM-IV TP...) et de Microsoft (MTS), c'est l'exécution des transactions en multi-threading avec du code partagé (donc complètement réentrant). Il peut s'avérer coûteux d'affecter un thread à chaque transaction en cours (ce nombre peut se monter à plusieurs centaines), si bien que l'on distingue les transactions actives des transactions dormantes, les premières constituant un cache de l'ensemble de toutes les transactions. Le nombre de transactions actives maximum est au moins égal à celui du rapport entre le temps d'attente des accès à la base de données au temps consommé par le processeur dans une transaction.

Intégrité

Un système transactionnel doit garantir que les effets extérieurs des transactions terminées aient leur intégrité garantie. Le client est en droit de revendiquer que ce qui lui a été vendu soit sa propriété. Ceci implique que les divers aléas informatiques n'aient pas d'autre impact que la détérioration du temps de service. Ces aléas avaient souvent jadis des causes matérielles (pannes du processeur ou de support des bases de données. Ils sont aujourd'hui des défauts d'intégrité soit du logiciel, ou des impasses de conception du logiciel d'application. Ils étaient et restent encore inhérents à la logique du système transactionnel où plusieurs clients peuvent être en train simultanément de rechercher le fond du stock de produits et le moins rapide à se décider peut avoir de mauvaises surprises. Il n'est d'ailleurs pas possible d'assurer la permanence du service lorsque la base données doit être dynamiquement mise à jour par un programme batch de renouvellement de stock ou d'extraction de stocks périssables. Il est nécessaire dans ce cas que le système transactionnel mime au mieux le comportement normal d'un processus humain.

Pour cela un mécanisme de contrôle de la concurrence entre threads doit être créé et il doit couvrir tous les accès aux bases de données y compris ceux du batch processing. En cas

Chapitre II : Historique des Systèmes d'information

d'étreinte fatale, le mécanisme doit tuer le thread le moins coûteux à recommencer. Bien entendu, il est nécessaire d'annuler toutes les modifications faites sur les bases de données avec un journal des modifications.

Commitment

la primitive la plus innovante d'un système transactionnel s'appelle la prise de commitment: "j'ai fini mon travail, vous pouvez relâcher tous les verrous que j'avais fermé". Cette primitive est implicite à la fin de la transaction, mais peut aussi se prendre en cours. Lorsqu'elle concerne des systèmes distribués il n'est pas infrequent que l'ordinateur terminal en est un commitment et doit se prendre en deux phases, l'une de préparation, l'autre de commitment effectif.

2.1.2.4 Affrontement du mode bloc/mode caractère

La mise en place de cette première vague de l'informatique réseau ne s'est pas faite sans querelles d'architectures. La question clé portait sur le mode de fonctionnement des terminaux et donc aussi sur le type de réseau de télécoms nécessaire. Deux modèles de fonctionnement des terminaux qui étaient alors en lice : le mode bloc et le mode caractère.

Le mode caractère :

Le terminal emblématique du mode caractère, c'est le VT100 de DEC introduit en août 1978 et qui est devenu par la suite le standard pour les terminaux en mode caractère. La configuration du VT100 était réalisée au moyen d'écrans interactifs affichés sur le terminal lui-même. Les paramètres étaient sauvegardés dans une mémoire non volatile. Le VT100 a été le premier terminal de DEC à être équipé d'un microprocesseur standard du marché, le 8080 d'Intel.

Le fonctionnement du mode caractère est facile à comprendre : à chaque fois que l'utilisateur frappe au clavier de son terminal, ce dernier envoie le caractère (d'où le nom du mode) entré à l'ordinateur central via la liaison réseau. Cet envoi systématique a pour conséquence qu'on demande beaucoup de transmissions de petites tailles au réseau en question. Quand il s'agissait de liaisons séries sur la distance de l'étage d'un immeuble, cela ne posait pas de problème. En revanche, sur des liaisons à grandes distances, les performances pâtissaient sensiblement à répétition du réseau.

Le mode bloc :

Si le mode caractère était omniprésent dans les minis, le mode bloc lui était la norme dans le domaine des mainframes car il était plus adapté aux transmissions longues distances : la saisie de l'utilisateur n'était transmise qu'à l'initiative de ce dernier et elle était envoyée complète. L'ordinateur central recevait cette image en bloc et comparait avec l'image d'avant qu'il avait gardé en mémoire afin d'en déduire les différences. Ce type de fonctionnement était conçu pour s'adapter aux moniteurs transactionnels de l'époque et était bien particulièrement au

Chapitre II : Historique des Systèmes d'information

découpage en trame des premiers réseaux informatiques. Le 3270 d'IBM était le terminal emblématique du mode bloc.

Ceci dit, le mode bloc s'il était économe en ressources techniques, imposait une ergonomie très frustrante : l'utilisateur n'avait droit à aucune aide ou aucun contrôle de saisie tant qu'il n'avait pas transmis sa transaction. Au contraire, les applications reposant sur le mode caractère pouvaient guider et réagir immédiatement aux saisies de l'utilisateur tout au long du processus, au niveau le plus fin et même en fonction du contexte.

A cause de la contrainte des réseaux longues distances, il était encore trop tôt dans l'évolution de la technique informatique pour que le confort de l'utilisateur prime, en mettant à sa disposition des interfaces capables de réagir immédiatement à la moindre saisie ou avec des langages évolués, simples à programmer. C'est donc fort logiquement que le mode bloc l'emporta dans la bataille qui l'opposa au mode caractère. En conséquence, toutes les initiatives qui avaient commencé à apparaître comme les L4G et la notion d'infocentre furent mise en sommeil.

2.1.2.5 Crise du logiciel

Après les deux premières vagues du système d'information, la plupart des applications de production et de traitement administratif sont développées et accessibles depuis des terminaux reliés au mainframe via des réseaux informatiques propriétaire et fermés. Ce premier niveau d'achèvement est spectaculaire pourtant, l'informatique souffre de son succès, on a appelé cette période "la crise du logiciel", car les utilisateurs, mis en appétit par ce nouveau visage de leur informatique, en demandaient toujours plus : plus d'applications et plus de terminaux. Le service informatique suit alors cette montée en puissance comme il peut. Le site central grossit, on embauche des spécialistes systèmes et réseau, etc.

[Ateliers de Génie Logiciel Christine Solnon 1996/97]

Qu'est ce qu'un logiciel ?

"Le logiciel est l'ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de l'information"
(arrêté du 22 déc. 1981)

Autrement dit, et de façon plus générale, un logiciel est un ensemble de programmes informatiques (du code) mais également un certain nombre de documents se rapportant à ces programmes et nécessaires à leur installation, utilisation, développement et maintenance: spécifications, schémas conceptuels, jeux de tests, mode d'emploi, ...

La "crise du logiciel" est apparue à la fin des années 60 et provient d'un décalage entre les progrès matériels d'une part et logiciels d'autre part: alors qu'apparaissaient les ordinateurs de la troisième génération, de plus en plus puissants et de moins en moins coûteux, la construction de logiciels restait dans le domaine de l'artisanat et du folklore, où chacun y allait

Chapitre II : Historique des Systèmes d'information

de sa petite recette. De fait, alors que les nouvelles machines rendaient possibles des applications jusqu'alors irréalisables, les méthodes de développement logiciel ne s'appliquaient pas à de grands systèmes: [Soln, 96/97].

- la construction de logiciels coûtait très cher (200 millions de dollars pour fabriquer OS-360),
- les délais n'étaient pas respectés (2 ans de retard pour les premiers compilateurs PL/1, Algol 68, ADA),
- les logiciels n'étaient pas évolutifs (parfois écrits en assembleur pour un type de machine) ce qui les rendait très rapidement obsolètes,
- avec des performances poussives (Univac, le système de réservation pour United Air Lines au début des années 70 n'a jamais servi car les temps de réponse étaient trop longs !),
- une fiabilité aléatoire (la sonde américaine qui devait aller sur Vénus s'est perdue, à cause d'une mauvaise instruction... plus récemment, la trajectoire de Ariane 5 a été modifiée à cause d'un débordement de capacité),
- et une convivialité discutable (des interfaces homme/machine inexistantes).

La crise du logiciel (software crisis) peut tout d'abord se percevoir à travers ses symptômes:

- Le coût de développement d'un logiciel est presque impossible à prévoir et le délai de livraison n'est que rarement respecté. On cite ainsi dans la littérature des dépassements moyens du coût budgété et du délai prévu respectivement de 70% et 50%.
- La qualité du logiciel livré est souvent déficiente. Le produit ne satisfait pas les besoins de l'utilisateur, il consomme plus de ressources que prévu et il est à l'origine de pannes.
- La maintenance du logiciel est difficile, coûteuse et souvent à l'origine de nouvelles erreurs. Mais en pratique, il est indispensable d'adapter les logiciels car leurs environnements d'utilisation changent et les besoins des utilisateurs évoluent.
- Il est rare qu'on puisse réutiliser un logiciel existant ou un de ses composants pour confectionner un nouveau système, même si celui-ci comporte des fonctions similaires. Tout amortissement sur plusieurs projets est ainsi rendu impossible.

La raison de fond de la crise du logiciel réside dans le fait qu'il est beaucoup plus difficile de créer des logiciels que le suggère notre intuition.

Comme les solutions informatiques sont essentiellement constituées de composants immatériels, tels des programmes, des données à traiter, des procédures et de la documentation, on sous-estime facilement leur complexité. Déjà la taille des programmes montre que cette complexité est souvent bien réelle: un million de lignes pour un logiciel de commande et de navigation d'un avion moderne, le décuple pour une station orbitale.

Chapitre II : Historique des Systèmes d'information

Pour maîtriser la complexité des systèmes logiciels, il convient de procéder selon une démarche bien définie, de se baser sur des principes et méthodes, et d'utiliser des outils performants. On arrive ainsi à gérer des projets complexes et à élaborer scientifiquement des solutions. Le génie logiciel (software engineering) cherche donc à établir et à utiliser des principes sains d'ingénierie dans le but de développer économiquement du logiciel qui est fiable et qui fonctionne efficacement sur des machines réelles.

Arrivé à ce point, il nous semble important de définir un certain nombre de notions inhérentes au génie logiciel.

Commençons par la notion de logiciel que nous avons défini aux précédents chapitres. L'Organisation internationale de normalisation, appelée brièvement ISO, a défini en 1981 le logiciel (software) comme une création intellectuelle rassemblant des programmes, des procédures, des règles et de la documentation utilisés pour faire fonctionner un système informatique. Pour renforcer le caractère intellectuel du logiciel, l'ISO précise en plus que le logiciel existe indépendamment des supports utilisés pour le transporter, ce qui sous-entend qu'il ne faut pas confondre le logiciel avec son support.

Le mot génie, utilisé en général accompagné d'un adjectif, comme dans génie civil, génie chimique ou génie atomique, désigne, d'après le Petit Robert, les connaissances et techniques de l'ingénieur. Ce terme est donc synonyme de science de l'ingénieur (engineering) qui peut être définie comme l'application des sciences et mathématiques par laquelle les propriétés de la matière et les sources d'énergie de la nature sont rendues utiles à l'homme dans des constructions, machines, produits, systèmes et procédés (traduit de l'anglais d'après le dictionnaire Webster).

Une première définition du génie logiciel consiste donc à dire qu'il s'agit de la science de l'ingénieur-informaticien spécialiste du logiciel. En explicitant cette définition, on peut encore affirmer que le génie logiciel est l'application des sciences et mathématiques par laquelle les possibilités d'un équipement informatique sont rendues utiles à l'homme à l'aide de programmes, de procédures, de règles et de la documentation associée. [4]

Remarquons maintenant que le mot anglais engineering possède par ailleurs un sens plus restrictif traduit en français par ingénierie, dont c'est la seule acception. D'après le Petit Robert, l'ingénierie est l'étude globale d'un projet industriel sous tous ses aspects (techniques, économiques, financiers et sociaux) coordonnant les études particulières de plusieurs équipes de spécialistes. Une deuxième définition possible, complémentaire de la première, du génie logiciel est donc la suivante: Le génie logiciel cherche à établir et à utiliser des principes sains d'ingénierie dans le but de développer économiquement du logiciel qui satisfait les besoins, qui est fiable et qui fonctionne efficacement sur des machines réelles [Fritz Bauer, 1969].

L'une et l'autre des définitions que nous venons de voir montrent que l'objectif premier du génie logiciel, comme d'ailleurs de toutes les sciences de l'ingénieur, n'est pas l'acquisition de nouvelles connaissances, mais plutôt leur mise en application, donc l'accumulation de savoir-

Chapitre II : Historique des Systèmes d'information

faire. Il en résulte que le génie logiciel ne constitue qu'une petite partie de l'informatique, mot intraduisible en anglais si ce n'est par *informatics*, puisqu'il inclut à la fois la science et la technique du traitement de l'information, alors que l'anglais utilise des termes spécifiques: *computer science*, d'une part, *computer engineering* et *software engineering*, d'autre part.

2.1.2.6 Echec des L4G : [Lef, 2001]

Dès qu'il est apparu que le développement du logiciel coûtait plus cher que le matériel, des consultants inventèrent la notion d'infocentre pour tenter de dissoudre le retard dans le développement de nouvelles applications par les utilisateurs eux-mêmes. Pour compléter la panoplie technique de l'infocentre, quelques innovateurs proposèrent la création des langages de quatrième génération, censés être bien plus simples à manipuler que les langages habituels (Fortran, PL1 et Cobol) des développeurs professionnels.

Initialement il s'agissait surtout de donner les moyens aux utilisateurs d'accéder aux bases de données pour produire des rapports ou pour soumettre des requêtes d'interrogations, sans passer par le service informatique de toute façon déjà débordé par les projets d'envergure et incapable de traiter rapidement des demandes simples et récurrentes. L'idée semblait excellente et devait permettre de satisfaire chacun : les utilisateurs, qui disposaient ainsi d'un peu d'autonomie et des moyens pour être plus réactifs, les informaticiens qui étaient déchargés de toute une série de demande de petits travaux répétitifs, souvent proches les uns des autres, très consommateur de temps et d'énergies. Il semblait que ces travaux étaient suffisamment simples pour être pris en charge par les utilisateurs sans demander trop d'assistance.

Les directeurs informatiques en adoptant cette généreuse recommandation faisaient coup double : ils répondaient à l'attente de leurs utilisateurs les plus influents sans surcharger leurs équipes d'études déjà dépassés. Hélas, les applications issues de ce compromis se sont révélées être de véritables gouffres à puissance (car les premiers L4G consommaient trop de ressources système) pour une faible satisfaction de la part de leur géniteurs/utilisateurs. Arrivé trop tôt avec des solutions techniques qui n'était pas en phase avec le contexte (les L4G exigeaient beaucoup de ressources système alors que celle-ci était encore rares et chères...), l'infocentre de dans sa première mouture fut généralement un échec.

A partir de là, on bascule clairement dans une autre époque et dans un autre rythme : c'est la fin d'un certain 'âge d'or' où l'informatique évolue rapidement certes, mais en restant plus au moins sur les mêmes bases. Le secteur connaissait une forte croissance mais restait tout de même 'un petit monde' où les vrais professionnels étaient encore relativement peu nombreux. Avec ce changement d'époque, le rythme change tout les cinq ans, une nouvelle mode apparaît qui prétend tout bouleverser et tout remettre en cause.

Souvent, cette nouvelle apporte des éléments positifs voir même nécessaire (le tout mainframe et la prééminence d'IBM avait un côté sclérosant qui ne fait que s'accroître avec les débuts des années quatre vingt). Mais aussi, quelque fois, on a assisté à un déballage

Chapitre II : Historique des Systèmes d'information

de 'grand n'importe quoi' et c'est l'irruption soudaine de l'intelligence artificielle qui a marqué le premier le 'premier épisode noir' de l'histoire de l'informatique.

2.1.2.7 Bref retour sur 'l'épisode noir' de l'IA :

Le terme 'd'intelligence artificielle ' remonte aux années soixante et attribué à Marvin Minsky du MIT, avec en particulier son article intitulé « Step Toward Artificiel Intelligence ».

On distingue trois périodes dans l'histoire de l'ITA : [Poul, 2011]

1. de 1952 à 1969 : c'est la période euphorique, on annonce le remplacement des experts humains par des systèmes experts.
2. De 1969 à 1988 : on commence à comprendre la difficulté de la tâche et le remplacement des systèmes humains par des systèmes expert est un échec (sauf dans quelques cas et bien spécifiques et particulièrement adaptés).
3. De 1988 à 1993, c'est l'hiver de l'IA : ralentissement des recherches en IA à cause des promesses non tenues.

C'est principalement au tour de la notion de 'système expert ' que le drame c'est noué.

Historiquement, le premier système expert fut Dendral en 1965, créé par les informaticiens Edward Feigenbaum, Bruce Buchanan, le médecin Joshua Lederberg et le chimiste Carl Djerrassi. Il permettait de d'identifier les constituants chimiques d'un matériau à partir de spectrométrie de masse et de résonance magnétique nucléaire. Mais le plus connu fut certainement Mycin en 1972-1973 : un système expert de diagnostic de maladies de sang avec un vrai moteur et une vraie base de règles. Les systèmes experts ont eu leur heure de gloire dans les années 1980. Les startups et les projets se sont multipliés et les systèmes experts allaient être partout, devaient être partout.

On découvrit bientôt que les bases théoriques n'étaient pas encore assez mûres pour réaliser toutes les attentes. Pour réussir dans de vrais problèmes de grande taille, il fallait résoudre des problèmes de base comme la logique non monotone et le raisonnement incertain. Or ces problèmes n'étaient pas très compris. Par conséquent, vers la fin des années 1980, le grand boom des systèmes expert toucha à sa fin et l'IA entra dans la phase du creux de désillusion de la courbe de Gartner (voir figure).

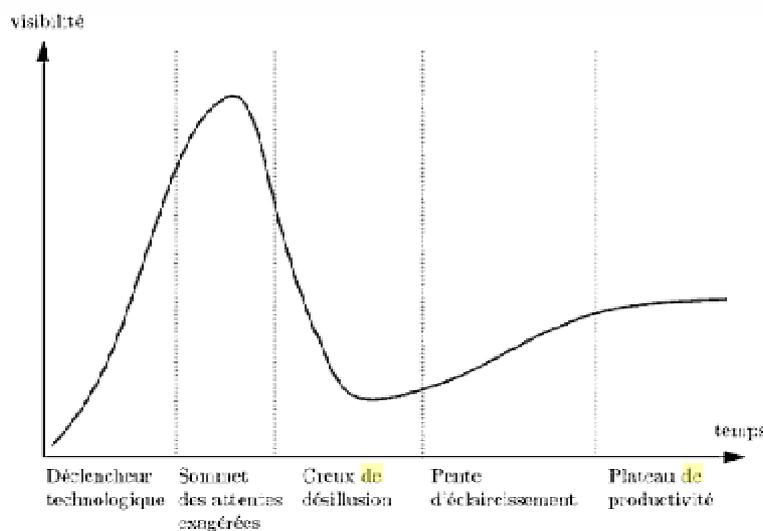


Figure 07 : la courbe hype [Falt, 2009]

Cette phase a conduit à l'hiver de l'IA, une période de déception qui a duré jusqu'en 1995 [Falt, 2009]. Certains sous-domaines sont devenus indépendants de l'IA. Le traitement de la langue et celui de la parole ont adopté des approches statistiques ne reposant plus sur des grammaires formelles développées par des linguistes, ce qui a conduit aux méthodes de recherche d'information utilisées dans les moteurs de recherche d'aujourd'hui. La vision par ordinateur est devenue un domaine en soi et a développé ses connections avec le traitement des signaux et des images.

Cependant, le boom des années 1980 a produit de nombreux chercheurs en IA, qui, pendant les années 1990, ont réalisés des progrès importants sur les bases théoriques qui posaient tant de problèmes pour les systèmes experts.

2.1.3 Troisième phase – l'utilisateur : document et Gui

Les progrès de l'ère du PC apportèrent un lot de nouvelles possibilités qui, fatalement, modifièrent considérablement le paysage informatique. Le système d'information ne fut pas épargné et c'est l'utilisateur individuel qui prit alors sa revanche sur l'approche applications collectives uniquement et avec des interfaces rustiques qui prévalait jusque-là. Tout d'un coup, le PC a permis et imposé « la bureautique ».

Cette dernière est une notion large qui couvre l'ensemble des applications permettant le traitement et la manipulation de tout type de document.

Chapitre II : Historique des Systèmes d'information

C'est l'ensemble des moyens et méthodes appliqués aux activités de *bureau* permettant de traiter informatiquement des informations écrites, visuelles ou sonores [6]. Le terme de « *bureautique* » est apparu pour la première fois en 1976 (en tant que traduction de l'expression anglaise *Office automation*), dans un discours de *Louis Naugès* intitulé « *les systèmes d'information numériques* ». Il faisait officiellement partie de la langue en décembre 1991 par arrêté du 22 décembre 1981 de la *Commission de Terminologie Informatique française*.

Le but de la bureautique est ainsi de fournir des éléments permettant de simplifier, d'améliorer et d'automatiser l'organisation des activités d'une entreprise ou d'un groupe de personnes (gestion des données administratives, synchronisation des rendez-vous, ...).

Etant donné que l'organisation de l'entreprise passe de plus en plus par un besoin en terme de communication, la bureautique ne s'arrête pas aujourd'hui à la simple saisie de notes manuscrites mais comprend notamment les activités suivantes :

- échange d'informations
- gestion des documents administratifs
- manipulation de données numériques
- planification de rendez-vous et gestion des emplois du temps
- un traitement de texte
- un tableur
- un outil de présentation (appelé parfois *préAO* ou *pre-AO*)
- une base de données

Les applications bureautiques sont destinées à être installées et utilisées sur PC. Elles visent directement l'utilisateur individuel par opposition à la masse des utilisateurs traité, de façon collective par les grandes applications de gestion de mainframe.

Pour les applications graphiques, elles ne se sont pas imposées immédiatement, il a fallu attendre quinze ans entre le lancement du Macintosh (1984) et celui de Microsoft Windows 3.0 (1990) pour constater que la lame de fond commençait à déborder du cadre initial de la bureautique.

Chapitre II : Historique des Systèmes d'information

2.1.3.1 Le revamping pour faire « comme si » :

2.1.3.1.1 Le client/serveur :

L'absence d'homogénéité entre les sources d'information locale et centrale donne lieu à un décalage entre les besoins exprimés par les utilisateurs et les services rendus par l'informatique classique ; il se crée donc une incohérence croissante des systèmes d'information.

Dans les années 90, la volonté de mettre en place un système informatique cohérent, exploitant au mieux les capacités de l'outil informatique, nécessite d'adopter une répartition plus équilibrée des traitements et des données entre le micro-ordinateur et les sites centraux et départementaux. Dans ce contexte, le client-serveur se présente comme le concept fédérateur des outils de traitement de l'information.

Le client-serveur proposait une architecture rationnelle permettant d'utiliser efficacement la puissance des ordinateurs centraux tout en bénéficiant du confort des graphiques des PC, mais aux prix d'une remise en cause radicale des développements existant, un saut que beaucoup d'entreprises refusèrent de faire dans un premier temps.

Alors pour faire comme si et concilier les deux mondes, une nouvelle mode apparut à la fin des années 80 : *le revamping*

Le revamping est d'après le dictionnaire du guide informatique [5] une technologie qui permet d'accéder aux anciennes transactions pour terminal passif des mainframes à l'aide d'une interface graphique sur micro (Windows, généralement). Le revamping qui est une vraie interface active (elle charge, voire contrôle, les zones de saisie), ne doit pas être confondue avec l'émulation qui ne fait qu'afficher l'écran de terminal passif à l'identique. Son évolution est le **web-to-host**.

Avec le revamping on se contentait d'habiller les écrans en mode texte des terminaux de l'application centralisé avec des dialogues en mode graphique.

A chaque map d'écran classique correspondait un dialogue ou une fenêtre gérée par l'interface graphique. L'habillage était réduit à sa simple expression car, même si une application graphique complète recouvrait l'application centralisé, la première n'était forcément que le reflet exact de la seconde : même découpage, même déroulement.

On s'est contenté d'intercepter le flux d'affichage venant de l'application centralisée, de faire le tri entre données utiles et éléments de décors et de placer les données utiles dans les fenêtres ou les dialogues de l'application graphique.

2.1.3.1.2 Faire du neuf avec du vieux :

Les outils du revamping se sont multipliés à la fin des années 80, ils avaient pour vocation d'offrir aux applications centralisées existantes un aspect graphique au moindre coût. Cette approche a connu une certaine vogue pendant quelques années, principalement parce que ces

Chapitre II : Historique des Systèmes d'information

outils minimisaient le travail de réécriture nécessaire à la prise en compte du mode graphique pour les applications centralisées, typiquement en mode bloc.

En dotant facilement ces applications d'une couche supplémentaire leur donnant un aspect graphique plus au goût du jour, on prolongeait leur durée de vie sans les modifier pour autant.

De plus, cela préservait un des arguments les plus importants en faveur des applications centralisées : leur fiabilité. Le revamping semblait présenter tous les avantages : le mode graphique sans effort et une sécurité de fonctionnement intacte puisque l'application restait telle quelle.

Pourtant, l'enthousiasme n'a pas duré et les promesses du revamping n'ont pas été tenues parce qu'il était déjà très difficile de concevoir correctement des applications nouvelles avec les contraintes spécifiques du mode graphique ; mais le lifting d'anciennes applications fonctionnant avec un enchaînement séquentiel des pages –écrans s'est révélé plus ardu encore.

2.1.3.1.3 Aller au-delà du revamping de surface :

Pour tenter de contourner cette limite, une forme évoluée de revamping apparut alors que : on ne se contenait plus de mettre en œuvre une réplique 'graphique' d'une application non graphique, on construisait une application entièrement nouvelle. Cette nouvelle application, développée avec un environnement permettant de capter et de détourner le flux d'affichage de l'application caractères, proposait un découpage et un déroulement qui pouvait éventuellement être entièrement différents de l'application d'origine.

On imagine facilement tout le potentiel offert par cette approche ; il en résultait une bien meilleure prise en compte de l'interface graphique qu'avec l'approche revamping simple, puisque'on pouvait alors véritablement utiliser tous les objets disponibles. De plus, c'est même le contenu fonctionnel qui pouvait être redéfini par ce biais, permettant ainsi de réorienter l'usage et l'emploi d'une application existante.

Inconvénient : le déroulement séquentiel est le facteur limitatif :

Toutefois, cette façon de faire n'était pas sans limites. Tout d'abord, même si on pouvait compiler des écrans ou les renvoyer en 'rafale', ceci ne permettait pas de répondre à tous les cas de figure car l'application d'origine, en raison de son déroulement séquentiel, imposait quand même sa rigidité et son découpage. Sans compter que l'effort requis pour réaliser une application de revamping, évoluée était sans commune mesure avec l'approche précédente, qui, elle, se prêtait bien à une automatisation.

On était bien en présence d'un véritable projet de développement en environnement graphique mais avec des contraintes inédites et quelques fois pénalisantes.

Le stade des modifications en profondeur :

Pas découragés, les partisans du revamping tentèrent de gravir l'ultime degré : contourner les problèmes du déroulement séquentiel avec des aménagements dans l'application classique

Chapitre II : Historique des Systèmes d'information

afin qu'elle se prête mieux à une exploitation en 'double' (alors que précédemment on s'interdisait de toucher à l'application centralisée, c'était d'ailleurs un des dogmes de base du revamping et même son principal avantage).

Ainsi, le découpage et/ou déroulement de l'application centralisée était subtilement modifiés pour tirer parti plus complètement du mode graphique et de ses avantages ou encore, minimiser le nombre d'allers et retours entre le poste de travail et le site central.

Cumul des inconvénients :

Cette forme extrême de revamping cumulait alors deux opérations de développement : une sur le poste de travail et une sur le site central. Cela faisait beaucoup pour un résultat qui allait forcément rester modeste par rapport à une vraie application client-serveur.

De plus, dans tous les cas de figure (revamping de surface ou modification en profondeur), la maintenance de ces applications 'à deux visages' s'est révélée être très pénible. Tout simplement parce qu'il a fallu la parfaite coordination entre l'application centralisée et sa contrepartie graphique sur le poste de travail de l'utilisateur. Une charge trop lourde pour un gain trop faible.

2.1.3.1.4 Le mode graphique, la chance du client/serveur :

Le modèle client-serveur s'est imposé au début des années 80, L'idée sous-jacente à la notion est celle de service. Une application informatique conforme au modèle client-serveur se décompose en deux entités logiques (on parle aussi de processus) : une première entité consommatrice de service, appelée le client, et une seconde entité, fournisseur de ce service, appelée le serveur. [Brau, 1995]

Cette décomposition logique est à la base même du concept client-serveur. Les deux entités applicatives communiquent entre elles par échange de messages de type question-réponse.

Ce modèle était le seul moyen abordable pour développer une nouvelle génération d'applications inspirée des innovations apportées par l'interface graphique, l'orientation utilisateur et la communication inter-applications. Les quelques expériences menées çà et là avec des outils comme dBase avaient convaincu les développeurs que les applications « réseau » qui reposaient intégralement sur le poste client n'était pas viables au-delà d'une petite dizaine de postes. D'un autre côté, comme le caractère graphique des applications étaient devenu incontournable et compte tenu de l'incapacité des sites centraux à gérer le graphisme, il fallait bien trouver un modèle de répartition des applications qui permette une mise en œuvre simple et efficace.

L'équation était simple : je ne peux pas rester centralisé à cause de Windows et je ne peux pas répartir les données de l'entreprise jusqu'aux postes de clients, je dois donc relier mes

Chapitre II : Historique des Systèmes d'information

applications hébergées et exécutées par les postes de travail aux bases de données manipulées et gérées par les grands systèmes.

Sel le client-serveur apportait à l'époque une réponse acceptable, en termes tant de performance que de facilité de mise en place. Et c'est ainsi qu'il s'est imposé, en accompagnant Windows dans sa généralisation et en lui permettant de ne pas rester confiné à la seule bureautique.

2.1.3.1.5 Le client/serveur de données domine

Au-delà du client-serveur, les possibilités étaient déjà nombreuses mais, en fait, le choix était simple : on pouvait surtout faire du client-serveur de données.

Le client-serveur était certainement la déclinaison la plus populaire dans le cadre du client serveur traditionnel ou départemental. Avec cette première forme de client-serveur, l'articulation entre client et serveur reposait entièrement sur le middleware du serveur qui abritait et gérait la base de données. La répartition des traitements était également fort simple : l'application résidait presque complètement sur le client qui faisait presque tout puisqu'il traitait en direct avec le serveur de données. Le rôle de ce dernier était d'exécuter les ordres SQL envoyés par les applications clientes. La base de données était donc accessible sans intermédiaire, les règles de gestion devaient être codées en son sein grâce à des mécanismes comme les mécanismes de déclenchement.

Cette déclinaison du client serveur s'est très rapidement et largement répandue, favorisée par l'essor des SGBD relationnels qui reposé dès leur origine sur ce modèle pour ce qui est de leur fonctionnement réparti. Tous les éditeurs de SGBDR ont commercialisé un middleware permettant d'accéder à leur produit à travers un réseau.

2.1.3.1.6 Le SGBDR s'implémente au cœur du système d'information :

L'une des conséquences dues au client-serveur concerne les SGBDR qui ont profité pour prendre place au cœur du système d'information. Le modèle relationnel cherchait lui aussi un vecteur qui lui permettait de s'imposer dans les entreprises, ce fut également le client-serveur. Car sur le plan technique, il semble bien que ce soit le modèle de type « client-serveur de données » qui se soit imposé, plus dans la réalité des sites que dans l'esprit des puristes.

Pourquoi cette victoire alors que la plupart des observateurs prévoyaient plutôt une généralisation de l'utilisation du « traitement coopératif » avec le recours à des protocoles sophistiqués comme APPC.

Parce que l'accès à des serveurs de données via des requêtes SQL était la solution la plus simple à développer et, tout aussi important, la plus facile à déployer. Les éditeurs de bases de données ont tous, très tôt, fournis le middleware adéquat pour relier les applications clientes à leurs serveurs, un middleware propriétaire, payant et ne permettant pas l'accès aux produits concurrents, mais opérationnel tout de suite sans programmation ni configuration complexe.

Chapitre II : Historique des Systèmes d'information

Le client-serveur n'a pas débouché sur des applications concrètes pour deux raisons :

1. Les applications client-serveur étaient difficiles à développer.
2. Les applications client-serveur étaient difficiles à déployer.

Ces applications étaient difficiles à développer car les environnements de développement étaient complètement nouveaux et s'appuyez sur un mode de programmation inédit jusque là : le mode événementiel. Ce type de développement était le mode naturel des applications s'exécutant en mode graphique mais ces projets exigeaient un véritable saut culturel aux programmeurs habitués au COBOL et à CISC.

Mais le plus difficile n'était pas de développer les applications client-serveur, car les environnements de développement graphique de qualités ne manquaient pas. Le vrai problème, c'était bien de mettre ces applications à disposition des utilisateurs en installant les parties clientes sur leurs postes de travail. Le déploiement des applications souffraient d'un mal qu'on qualifie de « syndrome des derniers kilomètres ». Ce qui coûtait cher et qui était délicat à maîtriser, c'était de devoir se déplacer jusqu'à la station de l'utilisateur pour installer les différents modules clients, le run-time, les DLL indispensables et de configurer les paramètres du middleware et autres incontournables de ce genre.

Les responsables informatiques des entreprises se sont vite aperçus que ce syndrome était bien le bien le principal obstacle à la généralisation des formes classiques du client-serveur.

On pourrait dire que le client-serveur dans sa première dans sa première forme s'est heurté sur le saut culturel de l'interface graphique et s'est brisé sur le mur du déploiement.

2.1.4 Quatrième phase – donner un sens aux données : le décisionnel

Les spécialistes du domaine avait imaginé que le client-serveur allait s'améliorer progressivement et permettre le développement d'applications de plus en plus ambitieuses. Le but ultime était bien évidemment de se positionner comme une alternatif du mainframe afin de permettre le rajeunissement du patrimoine applicatif.

Au lieu de faire sa percée sur le terrain des applications transactionnelles qui avait été largement investi par les mainframes, le client-serveur permit l'émergence d'un nouveau domaine fonctionnel : le décisionnel ou Business Intelligence.

Si le transactionnel permet de produire les informations détaillant le flux des affaires (commandes, bon de livraison, etc), le décisionnel permet de suivre cette production, d'avoir de la visibilité sur ces flux et donc de pouvoir piloter l'activité de l'entreprise. Le client-serveur avec ces outils clients graphique et ses serveurs de données représentait la plate forme idéal pour concrétiser enfin les idées déjà testées quelques années avant dans le cadre du concept d'infocentre.

Chapitre II : Historique des Systèmes d'information

Le décisionnel a connu un grand succès et s'est décliné, raffiné en différentes modes ayant datawarehouse puis datamining. Toutes ces applications permettaient enfin de faire parler les données, d'anticiper les tendances, de produire des prévisions et indicateurs qui, désormais, sont suivis régulièrement par les directions qui ne sauraient plus comment s'en passer.

2.1.4.1 Situation débloqué par le client-serveur :

D'abord du côté des moyens et des outils, les fameux langages de 4^{ème} génération, censés faciliter l'accès et la manipulation des données dans les bases n'étaient pas en phase avec leurs objectifs. En effet, ces derniers n'étaient pas réellement conçus pour être directement manipulés par de vrais utilisateurs.

Le problème, c'est que ces applications se révélaient également être des gouffres à puissance mettant à genoux les plus gros sites centraux dès qu'on les sollicitait de façon un peu intense.

De plus, les bases de données n'étaient pas adaptées à une interrogation en clair. Les SGBD relationnels qui étaient la bonne solution pour des interrogations à la volée n'en était alors qu'à leurs débuts. Et même avec des SGBDR, les structures des données des bases de production étaient étroitement liées aux applications qui les exploitaient, il aurait fallu entreprendre un gros travail de mise en forme pour les rendre exploitable en mode 'interrogation'.

Le client-serveur a débloquent cette situation en s'appuyant sur des stations de travail autonome où la puissance est abondante. Parallèlement, sur les serveurs, les SGBD se sont généralisés, proposant une structure plus propice à l'interrogation, avec des accès banalisés par des standards comme ODBC (Open DataBase Connectivity, un standard proposé par Microsoft au début des années 90 afin de simplifier l'interface entre applications clientes sous Windows et les bases de données du marché). C'est donc le client-serveur et le mode graphique qui ont permis de concrétiser les promesses de l'infocentre avec quinze ans de décalage.

2.1.4.2 De l'infocentre au DataWarehouse :

Il est significatif de constater que c'est le même acteur (IBM Canada) qui est à l'origine de l'idée de l'infocentre et qui a récidivé quelques années plus tard en proposant la notion « d'entrepôt de données » (DataWarehouse).

Cette notion contient quelques règles simples qui précisent comment organiser la base de données d'un infocentre.

Le but final de l'entrepôt de données est qu'il puisse « produire du sens à partir de données brutes »

Apparu au début des années 1990 et mis en avant par W. Inmon et par R. Kimball, des spécialistes du monde des bases de données : « Un système de DW organise et conserve les données nécessaires aux processus informationnels et analytiques dans une perspective de

Chapitre II : Historique des Systèmes d'information

long terme. Ce système correspond à un ensemble de données orientées selon un sujet, intégrées, évoluant dans le temps et non volatiles, qui a pour but l'aide aux processus de prise de décision de gestion. » [Inmon, 1996].

En outre, un DW intègre d'une manière systématique des métadonnées afin de fournir un référentiel unique à l'ensemble des utilisateurs.

Ce qui constitue alors l'originalité d'un système de DW réside dans le fait que les SAD (Systèmes d'aide à la décision) n'interrogent plus directement les bases transactionnelles. En effet, l'entrepôt de données est une base de données dont la structure est orientée vers la prise de décision et qui se positionne entre les applications transactionnelles et les SAD.

Le principal avantage qui a conduit à la généralisation de ce type d'architecture est qu'elle ne remplace pas l'architecture transactionnelle précédente, mais qu'elle s'insère dans un existant informatique. De plus cette architecture met clairement en lumière la différence qu'il peut exister entre une donnée et l'information qui en résulte après un processus d'interprétation. Ainsi, dans un système de DW, les décideurs ont accès aux mêmes données afin de construire leurs propres informations.

Ce n'est pas sous sa forme initial que le concept popularisé, mais seulement à partir du moment où, repris par d'autres acteurs, il a trouvé son nom de scène (IBM l'intégrait dans sa stratégie dénommé *Information Warehouse* mais c'est bien sous le vocabulaire Data Warehouse que la notion s'est répandue). Presque au même moment, EF.Codd (l'inventeur de la notion de base de données relationnel) publiait les douze règles du modèle OLAP (On Line Analytical Processing), censés définir les besoins du décisionnel.

Des progrès mais pas de changement :

Si, il y a eu des progrès mais rien qui ait permis de changer la situation initiale qui se résumait aux points suivants :

- le front-end n'était pas standard ;
- le middleware n'était pas standard ;
- la bande passante nécessaire était importante ;
- le développement d'application était complexe ;
- le déploiement d'application était une opération coûteuse et difficile à mener sur des grands sites et/ou sur des sites éclatés géographiquement.

2.1.5 Cinquième phase - l'intranet : informations et connaissances

Au milieu des années quatre-vingt-dix, une nouvelle mode prit tout le monde par surprise : l'Internet !

Chapitre II : Historique des Systèmes d'information

Pourtant, la pénétration de l'Internet dans l'informatique professionnelle s'est d'abord faite via le biais d'Intranet, c'est à dire l'utilisation des standards de l'internet au sein des réseaux d'entreprise.

2.1.5.1 L'intranet :

"Les réseaux intranet inaugurent une nouvelle génération de systèmes d'informations orientés non plus seulement sur les fonctions mais surtout sur l'organisation de l'entreprise" [Alin et al. 1996]. Cette citation met en évidence l'objectif de l'intranet, c'est la résolution des rapports complexes entre système d'information et organisation. Pour l'entreprise, cet objectif nécessite des résultats rapides dans un environnement où les frontières tendent à disparaître.

L'intranet a joué le rôle du maillon manquant dans la chaîne de publication assisté par ordinateurs.

Les tentatives été nombreuses pour diffuser des documents autrement que sur support papier mais toutes ont buté sur le caractère non standard à porté limitée, des solutions préconisées.

2.1.5.2 Le web débloque la situation :

L'avènement du web a permis de sortir de cette situation bloquée : soudain, il y avait une solution disponible, simple, standard et répandue à des millions d'exemplaires. Tout d'un coup l'évidence s'est imposée aux utilisateurs l'élément manquant de la chaîne de fabrication/diffusion de document était enfin trouvé. Cette adéquation entre longtemps insatisfait et sa solution surgit de nulle part explique largement le succès initial du web dans les réseaux d'entreprises. Succès grâce auquel on a commencé à parler de l'internet.

L'intranet a été l'occasion d'une grande vague d'équipement réseau au sein des entreprises.

Les usages de l'intranet sont les suivants :

- § publier des documents pour diffuser des informations ;
- § mettre en place des forums thématiques pour gérer et activer la communication ;
- § permettre l'accès à des bases de données en lecture et écriture ainsi qu'à des traitements applicatifs dans un mode client-serveur.

Même si c'est avec la diffusion d'information que la vague intranet a démarré, les utilisateurs ne se sont pas contentés de simplement publier des pages statiques. Et plus on se familiarisait avec les technologies du web, plus on s'apercevait que leurs applications pouvait servir à des fins allant bien au-delà du seul domaine documentaire.

2.1.5.3 Emergence et divergence du "Web client-serveur"

L'élargissement de l'usage de l'intranet était inéluctable, d'autant que comme plate forme pour les applications client-serveur, le Web avait de solides atouts à faire valoir. On peut citer

Chapitre II : Historique des Systèmes d'information

son caractère standard et son articulation multi-niveaux qui est la base même de son architecture.

Avec cette mutation, il est devenu clair que le Web pouvait être considéré comme une nouvelle forme du client-serveur. Ce changement de nature fut tellement rapide que des déclinaisons variées du "Web client-serveur" ont vu le jour, intégrant plus au moins profondément des technologies alors émergentes comme Java ou COBRA.

Tout comme le client-serveur traditionnel, les débuts du Web client-serveur ont été placés sous le signe de la simplicité. Mais le web client-serveur dans ces premières formes était ressenti par de nombreux experts du marché comme étant trop frustré pour être efficace. Les habitués du client-serveur avaient du mal à admettre les limitations dues au fonctionnement rigide du Protocol HTTP : pas de gestion de session et un découpage de contexte de client obligatoirement liés aux pages qui s'enchaînent.

2.1.5.4 Une alternative au Protocol HTTP :

Pour changer cet état de fait, la plupart des observateurs pensaient et déclaraient qu'il fallait trouver le moyen de se passer du HTTP tout en restant dans le cadre du navigateur web puisque tout le monde le désignait comme le client universel.

Le véritable élément qui a encouragé ces experts à vouloir une évolution radicale du Web client-serveur c'est java. Avec ce langage il devenait possible de télécharger, à partir des pages HTML dans le navigateur Web, de véritables petits programmes (applets). Ces derniers pouvaient faire office d'applications fondamentales capables de court-circuiter le Protocole HTTP pour utiliser à sa place des middlewares plus sophistiqués comme COBRA (avec IIOP) ou RMI.

Dans ce cadre le http ne sert plus que de Protocole d'amorce, permettant de charger la page faisant référence à l'applet qui va initialiser et piloter le dialogue client-serveur.

S'étant rapidement aperçue que l'exécution de java au sein du navigateur Web était assez restrictive, ses partisans préconisèrent ensuite d'utiliser java directement comme socle au sein de Windows pour des articulations client-serveur classique. Sois avec une articulation à deux niveaux où l'application java cliente adresse la base de données à travers une couche nommée JDBC, qui joue à peu près le même rôle qu'ODBC. Sois avec trois niveaux où une application cliente java invoque des méthodes d'un service applicatif, également en java, à travers RMI ou même CORBA ; ce service applicatif gère ensuite l'échange avec la base de données via JDBC.

Cette nouvelle approche du client-serveur avait même été baptisée « Java Computing » par ses nombreux partisans.

2.1.5.5 L'échec du java Computing :

Le Java Computing fut un échec mérité. Il y eut des raisons techniques solides à cet échec. Citons entre autres :

Chapitre II : Historique des Systèmes d'information

- § l'exigence en matière de bande passante de CORBA ou RMI par rapport à HTTP ;
- § l'exigence en matière de ressources systèmes de l'exécution de java côté client ;
- § l'immaturité des outils permettant de développer ces nouvelles applications ;
- § le caractère non standard des protocoles de communications employés par CORBA et RMI qui était de ce fait bloqué par le pare-feu.

Les partisans du Java Computing qui affirmait que java allait investir l'informatique via l'intranet se trompaient lourdement car leurs raisonnements s'appuyaient sur des postulats erronés tels que :

- § il est nécessaire de reproduire le fonctionnement des interfaces graphiques sur le Web, donc de remplacer le HTML par java pour pouvoir retrouver le comportement événementiel des applications.
- § L'infrastructure réseau de l'Internet va évoluer rapidement vers les hauts débits, on peut et on doit donc remplacer HTTP par IIOP ou autre pour pouvoir mettre en place des dialogues plus élaborés.

Croire qu'il fallait remplacer HTTP par un mécanisme sophistiqué, c'était nier les progrès de la version 1.1 du HTTP. C'était également ne pas comprendre que ce dont on a besoin, ce n'est pas de toujours plus de complexité mais, au contraire, de simplicité, pour atteindre enfin l'interopérabilité entre les systèmes.

2.1.5.6 L'intranet consacre l'interface utilisateur du web :

Depuis les années 80 jusqu'au milieu des années 90, le domaine de l'ergonomie des interfaces utilisateurs a été dominé par les interfaces graphique (GUI pour Graphical User Interface). Les concepteurs qui ont défini les règles régissant les premiers GUI cherchaient une alternative acceptable aux interfaces de type ligne de commande, comme Ms-Dos, avec des représentations graphique du système de gestion de fichiers local. Leur but était de rendre facile à gérer quelque MO de ressources locales.

Java n'était pas appelé à reproduire l'interface graphique à la place de l'interface du Web parce que justement, l'interface graphique n'est pas ce qui convient pour obtenir des applications intuitives et simples à utiliser. L'interface Hypertexte du Web a prouvé qu'elle offrait une solution qui plaisait à la grande masse des utilisateurs et qu'elle permettait de mettre en place de véritables applications.

Le Web a préfiguré une nouvelle manière de présenter les informations, manière axée davantage sur la navigation car le volume à représenter est quasi infini. La vague intranet du système d'information a donc fait passer progressivement de la notion de GUI à la notion NUI (N pour network).

Chapitre II : Historique des Systèmes d'information

2.1.6 Sixième phase – l'extranet : optimisation par l'entreprise élargie

L'automatisation de la production et des traitements de masse a représenté le premier cercle. Le deuxième cercle contient l'aide à la décision avec l'accès individuel aux données et informations. L'informatique aborde de nouveaux territoires justifiant consentis dans l'infrastructure des systèmes distribués. Le troisième cercle concrétise le rêve de l'entreprise élargie, ce dernier est réalisé grâce aux réseaux.

2.1.6.1 Du système d'information aux systèmes de communication :

C'est un système qui n'est pas seulement constitué d'un serveur web et d'une messagerie aux normes Internet ; en fait, c'est la plupart des applications qui doivent pouvoir être utilisées depuis l'extérieur via l'Internet. Ce sont la plupart des bases de données qui doivent être en mesure de recevoir ou d'émettre des informations depuis un format standard.

2.1.6.2 Le traumatisme de l'an 2000 :

Justement, c'est bien la perspective du passage à l'an 2000 qui a poussé l'ensemble des entreprises à nettoyer les applications du système d'informations.

En quoi consiste le bogue de l'an 2000? [Dub, 99]

Le problème de l'an 2000 résulte du fait que certaines composantes informatiques utilisaient jusqu'à tout récemment deux caractères au lieu de quatre pour représenter l'année dans une information de date. Par exemple, le 1^{er} janvier 2000 serait représenté par 00/01/01 au lieu de 2000/01/01. Dans plusieurs cas, des mécanismes dotés d'une horloge électronique, comme les ordinateurs, interpréteront cette date comme 1900 au lieu de 2000. Certains mécanismes ne seront pas fonctionnels parce qu'ils interpréteront 00/01/01 comme étant « Aucune date n'est disponible ». S'ajoute au passage à l'an 2000 l'année bissextile que certains logiciels et systèmes d'exploitation ont omis de considérer. Un autre cas particulier est celui du 9/9/99 et le 99^e jour de 1999 où certains logiciels pourraient comporter des limites artificielles de programmation, c'est-à-dire qu'une série consécutive de « 9 » peut parfois être utilisée pour représenter la fin logique d'un algorithme quelconque. Par conséquent, bon nombre de composantes informatiques pourraient provoquer des erreurs de calcul, de tri, d'affichage et parfois même des arrêts complets.

On a déjà relevé divers problèmes dans un certain nombre de systèmes:

- les ordinateurs des usines de transformation des produits alimentaires et des entrepôts ont commandé des tonnes de nourriture qui seront détruites après avoir calculé que la période au cours de laquelle ces denrées pouvaient être consommées sans danger est révolue depuis près de 100 ans (en 1900);

Chapitre II : Historique des Systèmes d'information

- les enregistreurs de caisse des supermarchés ont refusé des achats avec des cartes de crédit qui auraient expiré il y a un siècle.

Il y a de gros risques que lors du passage au prochain millénaire, des problèmes similaires surgissent simultanément dans le monde entier et compromettent les transactions individuelles et les systèmes internationaux.

L'origine du problème remonte au début de l'ère informatique alors que le format abrégé des dates permettait de réaliser d'importantes économies sur les coûts d'utilisation de l'espace mémoire vive et de la mémoire sur support magnétique. Ce souci d'économie était empreint d'ignorance, ou du moins d'insouciance, quant aux conséquences futures. On présumait que les programmes et langages primitifs seraient remplacés bien avant le nouveau millénaire.

Tel ne fut pas le cas. Nombre d'organisations continuent à utiliser des systèmes informatiques mis au point il y a plus de 30 ans. De plus, malgré une baisse substantielle des coûts d'utilisation de la mémoire, les mauvaises habitudes se sont perpétuées sans se préoccuper outre mesure des répercussions. Ce n'est que vers la fin des années 80 que l'industrie informatique a sonné l'alarme et, malgré cela, plus d'une organisation a tardé à réagir pour maintes et maintes raisons. D'ailleurs, certains fabricants proposaient jusqu'à tout récemment du matériel non conforme à l'an 2000 tel que la génération des microprocesseurs 286, 386 et 486.

Même Windows 95 et Internet Explorer 4.0, pour ne nommer que ceux-là, comportent des problèmes reliés à l'an 2000.

2.1.6.3 Défis des systèmes d'information d'aujourd'hui :

Après le passage à l'an 2000, les systèmes d'information (SI) des sociétés ont été repensés sur le plan fonctionnel, et des pôles d'urbanisme ont été créés dans les grands groupes. « Avec la complexification des SI, le métier continuera de se développer, pronostique Christophe Longépé, directeur de l'urbanisme et des référentiels du Groupe Société Générale.

Pour résumer, les DSI sont dans la situation suivante : il n'est plus possible de faire évoluer le système sans le reconstruire et refondre le système devient impossible pour des raisons de : [Leva, 2002]

- complexité : « la complexité fait qu'on a du mal à en contrôler l'avancement, et à réunir les équipes compétentes nécessaires, ce qui oblige à faire largement appel à la sous-traitance et rend plus problématique encore la maîtrise globale. » [Leva, 2002]

- délais

- coûts

Les enjeux pour les DSI sont alors les suivants :

- Comment continuer de faire évoluer le système dans la maîtrise des coûts et de délais?

Chapitre II : Historique des Systèmes d'information

- Comment le faire en concentrant l'effort sur les parties à plus forte valeur ajoutée ?
- Comment, dans le même temps se libérer du joug des gros constructeurs ?

Devant l'ampleur des budgets attribués aux directions informatiques malgré la récession économique, les préoccupations des directions stratégiques se sont positionnées plus tardivement ailleurs. L'informatique coûte cher. La problématique est différente :

Ne pourrait-on pas transformer le système, certes dans la maîtrise des coûts et des délais, mais aussi dans une perspective à moyen terme voire à long terme, indépendante de l'obsolescence rapide des technologies, transformation accompagnant le plan d'entreprise ?

Ne faudrait-il pas que les informaticiens « passent d'une mentalité de Nomades à une mentalité de bâtisseurs ? » [Leva, 2002].

L'urbanisme des SI apparaît dès lors que la question suivante est posée : comment peut-on isoler ce qu'il faut refondre, ce qu'il faut maintenir et ce qu'il faut garder en l'état ?

Conclusion :

Pour conclure, nous pourrions dire que durant ces trente dernières années, l'évolution des systèmes d'informations nous a donné des enseignements qu'il serait bon de retenir pour l'avenir.

Mais ce champ des Systèmes d'Information qui est poussé par la technologie, n'a pas encore vraiment construit son corps théorique spécifique [Rodh ,2010], ce qui questionne encore son identité mais offre toujours une grande richesse et une forte attractivité.

Nous avons donc dans ce présent chapitre pu retracer l'historique de ces systèmes d'information en passant par les différentes mutations ainsi que les événements les plus marquants de cette discipline.

Le système d'information s'inscrit aujourd'hui dans une dimension stratégique de l'entreprise, qui consiste en son urbanisation. C'est ce qu' nous allons essayer de développer dans le chapitre qui va suivre.

Chapitre III

Urbanisation des Systèmes d'Information

Chapitre III : Urbanisation des Systèmes d'information

1. Introduction :

Malgré une indéniable amélioration dans l'utilisation de l'informatique, les entreprises ont encore du mal à concevoir une évolution de leur système d'information cohérente avec leurs priorités. Or l'un des critères de qualité d'un système d'information est son évolutivité, c'est-à-dire la souplesse avec laquelle il saura répondre tant aux évolutions du métier lui-même qu'à celles des solutions informatiques disponibles. Aussi, le thème de ce mémoire porte sur les systèmes d'information, et plus particulièrement sur l'urbanisation de ces derniers, peut être un support pour les projets d'évolution du système d'information. En effet, l'urbanisation est une discipline s'intéressant à l'évolution des systèmes d'information et qui ambitionne de doter les organisations de systèmes d'information adaptatifs, pour les mettre plus complètement au service de la stratégie.

Dans une première partie, les concepts clés de l'évolution du système d'information y seront développés : à savoir les notions d'architecture et d'urbanisation des systèmes d'information. La seconde partie sera consacrée à la notion de cartographie : les différents types de cartographie et les problématiques auxquelles ils s'adressent y seront présentés. Enfin, la troisième partie, traitera l'intérêt des cartographies pour le pilotage de l'évolution du SI.

2. La problématique du système d'information

2.1 Informatisation des entreprises et systèmes d'information

L'évolution des systèmes d'information depuis les années 1960 met en valeur des mutations profondes ayant fait évoluer les besoins des entreprises en matière de gestion des systèmes d'information. L'apparition même du terme de système d'information est révélatrice des transformations de mentalités : de cela découle l'émergence de nouveaux acteurs et de nouvelles entités de gestion des systèmes d'information. Apparaît notamment la problématique de l'évolution du système d'information qui doit être pilotée et non subie.

La complexité des systèmes d'information les rend de moins en moins flexibles dans un contexte économique où flexibilité, adaptabilité, interopérabilité sont les maîtres mots de la survie des entreprises. De quoi sont composés les systèmes d'information aujourd'hui ? Comment est-on passé de la notion de service informatique à la notion de direction des systèmes d'information ? Et enfin, pourquoi cette transformation a-t-elle impliqué un besoin de pilotage de l'évolution du système d'information ?

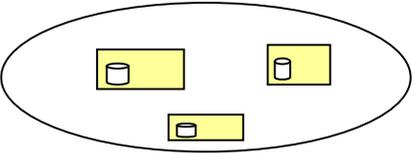
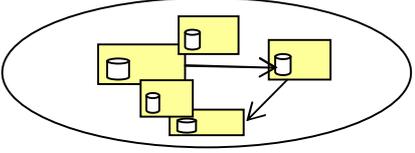
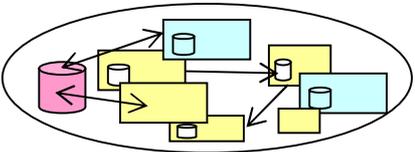
Chapitre III : Urbanisation des Systèmes d'information

2.2 Bilan de l'évolution

2.2.1 Un système d'information sédimenté

L'étude de l'histoire des systèmes d'information montre que dans de nombreuses entreprises, le développement des systèmes d'information a été subi plutôt que conduit. Sous la pression de l'urgence, on a réalisé des applications informatiques centrées sur des objectifs locaux et immédiats, sans grand souci de cohérence globale. L'évolution des moyens en matériel, logiciel et personnes n'a que rarement fait l'objet d'une réflexion approfondie et a le plus souvent été dictée par la pression des offreurs et les besoins à court terme.

Finalement, le SI des entreprises s'est développé de façon anarchique pour devenir un assemblage opérationnel mais fragile car non structuré. La figure, ci-après, montre l'évolution des SI au cours des dernières décennies et met en avant l'absence de structure auquel l'empilement d'applications a conduit.

	<p><i>Décennie 1960</i></p> <p>Quelques applications.</p> <p>Les données sont gérées par chaque application.</p>
	<p><i>Décennie 1970</i></p> <p>Quelques applications à quelques dizaines.</p> <p>Liens entre ces applications par partage de données.</p>
	<p><i>Décennie 1980</i></p> <p>Quelques centaines d'applications.</p> <p>Des liens plus nombreux entre applications et apparition de logiciels.</p>

Chapitre III : Urbanisation des Systèmes d'information

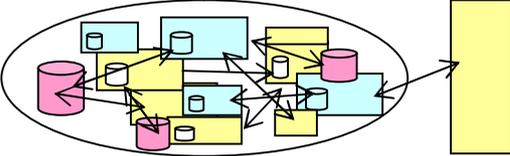
	<p><i>Décennie 1990</i></p> <p>Quelques centaines d'applications à quelques milliers.</p> <p>Beaucoup de progiciels, un SI tourné vers l'extérieur, avec parfois des parties externalisées.</p>

Figure 08 - Evolution des systèmes d'information [Dub, 02]

Ainsi, de nombreuses entreprises se trouvent confrontées à des systèmes d'information se caractérisant par des applications juxtaposées, des complications inutiles, une sémantique désordonnée et un désordre technique. Ainsi, bien qu'on parle de système, il n'y a pas vraiment de système puisque ce dernier se caractérise par son organisation et qu'on se trouve au contraire face à un ensemble désordonné d'applications de plus en plus nombreuses.

Les deux principaux problèmes auxquels on se trouve alors confrontés sont les suivants :

- les coûts de maintenance du système sont importants et couvrent une grande partie du budget informatique de l'entreprise,
- le système est peu évolutif (pour faire évoluer une partie, on est souvent obligé d'en repenser plusieurs autres), ce qui engendre des coûts importants (en temps et en argent).

2.2.2 Un besoin de cohérence

Face à au développement croissant des systèmes d'information, le besoin de cohérence apparaît évident. Ce n'est qu'au cours de la décennie 1980 que s'est imposée la nécessité d'une gestion stratégique des systèmes d'information, afin de permettre une cohérence entre le métier, le système d'information et le système informatique. Il s'agit du leitmotiv de l'ouvrage de Gérard Jean sur l'urbanisation du business et des systèmes d'information. Ce dernier préconise ce qu'il appelle l'"alignement stratégique", qui consiste à mettre en cohérence stratégie, processus, système d'information et système informatique. Les processus sont la traduction dynamique du métier et le pont entre métier et système informatique est fait grâce au système d'information.

La gestion stratégique du système d'information doit apporter la réponse à deux questions fondamentales :

Chapitre III : Urbanisation des Systèmes d'information

- Des systèmes d'information pour quoi faire ?

Il s'agit ici de faire le lien entre le système d'information et le métier. Quels applicatifs sont dédiés à quels métiers ?

- Des systèmes d'information organisés comment ?

Il s'agit ici de faire le lien avec le système informatique, en déterminant notamment comment dialoguent entre eux les différents applicatifs.

Pour répondre à ces deux questions, la notion d'architecture, déclinée sur plusieurs niveaux, a été reprise au profit des systèmes d'information. Cette notion est définie et développée dans le paragraphe suivant.

2.2 Les notions d'architectures

2.2.1 Analogies et parallélismes

2.2.1.1 *Le projet informatique comme un chantier*

De plus en plus souvent, on assiste en informatique à des analogies et à des parallélismes avec d'autres secteurs. L'informatique étant relativement jeune, elle commence à tirer profit des expériences des autres domaines, et notamment de celui du bâtiment.

En effet, l'émergence des technologies objets et de la programmation modulaire a fait apparaître une plus grande spécialisation des acteurs des projets informatiques. Si auparavant une nouvelle application était développée de façon 'artisanale', il apparaît de plus en plus nécessaire de structurer et d'organiser les développements. Différents acteurs apparaissent, chacun avec sa spécialité. Ainsi, on aura des spécialistes pour mettre en place le modèle de données, des spécialistes des composants techniques, des spécialistes des composants métiers, des spécialistes IHM... L'analogie est aisée à faire avec le secteur du bâtiment et la notion de chantier où interagissent le maçon, le charpentier, le couvreur, le peintre... Le chef de projet est alors chef de chantier et l'entité coordinatrice est à définir.

2.1.1.2 *L'architecte coordinateur*

Ce besoin d'entité coordinatrice a fait émerger la notion d'architecte¹ qui, dans le secteur du bâtiment joue ce rôle de coordination. Cette analogie peut se faire tant au niveau système informatique qu'au niveau système d'information.

¹ Le terme d'architecte vient du grec Arkhitektôn qui signifie maître constructeur. La définition qu'en donne le Larousse est la suivante : "personne qui conçoit le parti la réalisation et l'adéquation d'édifices de tous ordres et en dirige l'exécution".

Chapitre III : Urbanisation des Systèmes d'information

Ainsi, "le maître d'ouvrage du système d'information joue un rôle analogue à celui de l'architecte qui construit un bâtiment pour un client : le porteur légitime du besoin est le client, et toute la construction du bâtiment doit être orientée par l'ambition de le satisfaire." [Volle 00].

L'architecte devra ensuite veiller à la bonne conduite du chantier, sur lequel interviennent des corps de métier aux compétences diverses. C'est à lui de disposer l'architecture de façon à offrir des services de bonne qualité.

2.2.2 Définitions

La notion d'architecture a ainsi été adoptée pour désigner la structure de tout ou partie d'un système informatique. Cependant, la notion d'architecture en informatique est vaste, elle peut également représenter "un ensemble de règles de fonctionnement et de conception". [Martin99]. Au niveau du système informatique, on parlera d'architecture technique et applicative. Qu'en est-il au niveau système d'information ? A ce niveau, on parlera plutôt de l'architecture fonctionnelle qui traduit le métier dans le système d'information. Ce métier trouvant lui même sa structure dans une architecture métier.

Ainsi, l'architecture de système d'information est une discipline multiforme et transversale qui traite de "comment concevoir et construire un SI". Son objectif est d'anticiper les problèmes de faisabilité et de cohérence du futur système en fonctionnement, tout au long du processus logiciel, selon principalement 3 axes d'étude :

- la recherche de la vision globale,
- la construction "sur plan" et non sous l'unique pression de la demande,
- la notion de cible : il faut savoir où l'on va.

Cependant, on a vu que l'analogie avec l'architecture pouvait se décliner à plusieurs niveaux de description, depuis la notion d'architecture technique à la notion d'architecture métier en passant par les architectures applicative et fonctionnelle. Les définitions ci-après proposent une approche systémique* des notions d'architecture, partant de la vue métier pour arriver à la vue technique.

- La vue métier représente le métier de l'entreprise concernée (quels processus et quelles procédures ?)
- La vue fonctionnelle sera associée aux grandes fonctions de l'entreprise (que représente le SI ?).

Chapitre III : Urbanisation des Systèmes d'information

- La vue applicative exprime comment les fonctions sont réalisées (avec quelles applications ?)
- La vue technique détaille les moyens (avec quoi fait-on fonctionner les applications répondant aux fonctions du métier ?)

2.2.2.1 L'architecture métier

L'architecture métier représente "la structuration du système d'information par les activités métier de l'entreprise ou de l'organisme vis-à-vis de ses processus." [Longépé 01 (1)] En général, cette structuration se fait à partir des processus métier, décomposés en activités puis en procédures.

2.2.2.2 L'architecture fonctionnelle

L'architecture fonctionnelle est l'organisation et la structure des fonctions métier de l'entreprise. Ces fonctions sont identifiées et leur organisation apparaît *via* les échanges entre elles. La structuration se fait par la définition de blocs fonctionnels plus ou moins importants, qui communiquent entre eux.

L'architecture fonctionnelle pose les règles de gestions qui sont implémentées dans les applications. Ces applications sont structurées par une architecture applicative.

2.2.2.3 L'architecture applicative

L'architecture applicative définit les différents blocs applicatifs de l'entreprise, leur structuration et la façon dont ils communiquent entre eux. L'architecture applicative est donc la "description et l'organisation des applications informatiques (données et traitements) ainsi que des messages échangés par ces applications." [Longépé 01 (1)] Les applications sont regroupées dans des structures plus ou moins importantes : zone, quartier, îlot.

Il existe deux types d'architectures applicatives. L'architecture applicative inter-applications et l'architecture applicative intra-application [Gaye 00]. La première s'attache à détailler la structuration des applications en blocs et les échanges entre ces blocs et entre les applications. La seconde détaille les échanges au sein même d'une application. A ce niveau, il arrive parfois qu'elle soit confondue avec l'architecture technique : dans la pratique, elle se situe à la frontière entre architecture technique et architecture applicative. En effet, les règles relatives à la constitution des applications sont définies dans le cadre de l'architecture technique.

Chapitre III : Urbanisation des Systèmes d'information

1.2.2.4 L'architecture technique

L'architecture technique quant à elle, est une notion antérieure aux précédentes. En effet, les services informatiques responsables de leurs applications ont rapidement saisi la nécessité de structurer les ressources et les moyens matériels utilisés pour l'informatisation des activités de l'entreprise. L'architecture technique est donc "la description et l'organisation des différents moyens matériels (central, serveur, poste...), des logiciels de base (système d'exploitation, SGBD...) ainsi que des moyens de communication entre elles (réseaux)." [Longépé 01 (1)]

2.2.3 Cohérence des architectures

Comment définir ces différentes architectures de façon cohérente ? L'architecture applicative doit être supportée par une architecture technique adaptée et doit être cohérente avec les architectures métier et fonctionnelle. Quels sont les moyens dont l'entreprise dispose pour assurer cette cohérence globale ? Si l'on reprend le parallélisme réalisé au début de ce paragraphe, on peut se demander quel est l'acteur qui va s'attacher à assurer la cohérence des architectures des bâtiments qui nous entourent ? On en vient naturellement à penser à l'urbaniste. C'est ainsi que les concepts liés à l'urbanisme ont été repris au profit des systèmes d'information.

La plupart du temps, les entreprises disposent d'une architecture technique bien définie par les équipes techniques, mais il n'y a pas d'existant en termes d'architectures métier, fonctionnelle et applicative. Ces trois architectures vont découler d'une démarche d'urbanisme. Il faut définir des architectures qui représentent les cibles à atteindre et s'efforcer d'y tendre par un management efficace du système d'information dans sa globalité. Ce management efficace se traduit par une prise en compte de la stratégie de l'entreprise pour la définition de la cible. En outre, les architectures cibles à atteindre ne sont pas statiques et le système d'information doit être capable d'évoluer au même rythme que l'entreprise qui peut être bouleversée par des fusions remettant en question l'ensemble de son système d'information. Les cibles doivent être en cohérence avec un plan d'évolution global qui se traduit dans l'urbanisme. Ainsi, définir une architecture technique cible ne suffit pas : celle-ci doit être mise en parallèle avec une architecture fonctionnelle cible. L'ensemble fournira une cible opérationnelle.

Le rôle de l'urbanisation est de définir cette cible et de faire en sorte d'y tendre en entamant une rénovation progressive du système d'information existant, sédimenté comme le montre la figure 8.

Chapitre III : Urbanisation des Systèmes d'information

2.3 Le concept d'urbanisation des systèmes d'information

Initialement développé par Jacques Sassoan, le concept d'urbanisation des systèmes d'information séduit de plus en plus les grandes entreprises, et notamment les banques. Ces grosses structures "consommatrices" des systèmes d'information se dotent de plus en plus de cellules d'architectes et d'urbanistes. La période pionnière où le concept se trouvait confronté à de nombreux scepticismes est passée et l'idée a aujourd'hui fait son chemin.

2.3.1 La métaphore de la ville

La notion de cohérence globale des architectures amène à un autre parallèle que celui réalisé avec l'architecture : celui de la ville. En effet, si l'on considère le système d'information comme un assemblage d'éléments ayant chacun leur architecture propre et communiquant entre eux, pourquoi ne pas considérer le SI comme une ville ? Cette métaphore est issue des travaux de Jacques Sassoan [Sassoan 98], elle a été par la suite enrichie par Gérard Jean, président de la société de conseil Altime.

Il apparaît donc que le parallèle est aisé entre le monde "réel" des villes et le monde "virtuel" des systèmes d'information : à l'instar d'une ville, le SI se heurte à des problèmes de maîtrise de la complexité lorsqu'il atteint une certaine taille. En effet, dans la plupart des entreprises et organismes, le SI a été créé par les projets qui se sont succédé et qui étaient à la fois marqués par les besoins et les technologies d'une époque. Cette situation est similaire à celle d'une cité où les constructions successives expliquent le paysage actuel.

"Les systèmes d'information doivent évoluer à l'image de la ville où le changement est rarement brutal mais diffus ou par quartier, nos SI évoluent progressivement, sans remise en cause, en acceptant la cohabitation harmonieuse de programmes ou logiciels d'origines différentes et d'époques différentes"

Ainsi, dans la partie sur l'architecture nous avons assimilé une application informatique à un bâtiment. L'agencement d'applications contribuant à la construction du système d'information peut alors être assimilé à une ville. Par ailleurs, les problématiques rencontrées par le SI lorsqu'il croît sont similaires à celles rencontrées par la ville lorsqu'elle atteint une certaine taille. Celle-ci se trouve en effet confrontée à "une très grande variété de besoins, individuels et collectifs, opérationnels ou politiques qui, s'ils ne sont pas canalisés, peuvent conduire à des problèmes de fonctionnement et d'évolution : insalubrité, sécurité, congestion de la circulation automobile, dégradation de l'environnement architectural, dépassement budgétaires..." [Jean 00]

Pour le système d'information, les problématiques sont similaires :

Chapitre III : Urbanisation des Systèmes d'information

- variété des besoins, à différents niveaux²,
- besoins opérationnels et décisionnels,
- risque de vieillissement des applications,
- problèmes de sécurité des informations,
- saturation des réseaux,
- coûts importants pour la rénovation.

La solution proposée pour pallier à ces problèmes dans les villes est l'urbanisme. Qu'en est-il de l'application de cette solution au système d'information ?

1.3.2 Définition

1.3.2.1 Urbanisme de la cité

Commençons par définir l'urbanisme indépendamment du contexte des systèmes d'information. L'urbanisme est l'action réfléchie visant à aménager ou à restructurer physiquement et socialement l'espace (urbain et rural) en vue d'assurer l'unification la plus harmonieuse et la plus efficace des fonctions que remplit un site donné, singulièrement l'habitation et la circulation. Il est inséparablement une théorie et une pratique dont l'exercice entraîne le recours à une technique.

Dans son ouvrage sur l'urbanisme [Merlin 91], Pierre Merlin insiste sur le fait que l'urbanisme n'est ni une science, ni une technique, ni un art, mais plutôt une action réfléchie, une pratique. Cependant, l'urbanisme s'est trouvé assimilé à l'une ou l'autre de ces notions par les différents courants de pensée sur l'urbanisme. Ainsi, il apparaît que même dans sa définition première, la notion d'urbanisme est sujette à controverse : "L'urbanisme n'est pas aisé à définir" [Merlin 91]. Nous verrons par la suite que la cartographie est quand à elle un outil pour l'urbanisme et s'avère à la fois scientifique, technique et artistique.

1.3.2.2 Urbanisme des systèmes d'information

Ecrivons cette définition au profit des systèmes d'information : il s'agit alors d'entamer une action réfléchie visant à organiser ou restructurer aux niveaux physique et logique le système d'information en vue d'assurer une meilleure cohérence et une plus grande efficacité des

² On distingue quatre niveaux de besoin : individuel (c'est le cas par exemple de certains tableaux de bord), collectif (s'adressant à un service, un groupe d'utilisateurs), organisationnel (lorsqu'il affecte toute l'entreprise, comme la messagerie par exemple), et enfin inter-organisation (lorsqu'il concerne les échanges entre entreprises : c'est le cas de l'EDI par exemple) [Reix 00]

Chapitre III : Urbanisation des Systèmes d'information

fonctions que remplissent chacune des parties du système d'information, à savoir le traitement de l'information et sa communication. Il est simultanément une théorie et une pratique dont l'exercice entraîne le recours à une technique.

Si l'urbanisme des systèmes d'information est un concept franco-français essentiellement répandu dans l'informatique bancaire et financière, il commence à se répandre outre-atlantique : le Gartner Group le présente en effet aux USA dans la mouvance EAI - Enterprise Applications Integration avec le terme "city planning " en anglais.

Le fondateur du concept, Jacques Sassoon définit l'urbanisme des systèmes d'information de la façon suivante :

L'urbanisme des systèmes d'information est la technique de l'organisation des activités exercées par ces systèmes d'information [sassoon 98].

Il s'agit donc d'organiser le SI de façon à le simplifier. Urbaniser le système d'information consiste donc à trouver un découpage de ce système d'information et des principes de construction qui permettront :

- de le faire évoluer au même rythme que la stratégie et l'organisation,
- d'y intégrer rapidement de nouvelles technologies.

1.3.2.3 Architecture et urbanisme

En ouverture de *Manière de penser l'urbanisme*, publié en 1946, Le Corbusier³ soutient que "l'urbaniste n'est pas autre chose que l'architecte". Cependant, l'ouverture de l'urbanisme à la sociologie et à l'économie notamment, contraint à remettre en cause un tel formalisme. Dans une étude intitulée *L'urbanisme d'aujourd'hui : mythes et réalités*, le sociologue Henry Lefebvre soutient que "l'architecture et l'urbanisme doivent être distingués avec soin en tant qu'ils se situent à deux niveaux de la réalité sociale" : l'architecture est au niveau micro-sociologique, tandis que l'urbanisme est un problème macro-sociologique, posé au niveau de la société dans son ensemble.

De la même façon, il convient de bien distinguer dans le cadre de l'urbanisme des systèmes d'information, l'urbaniste de l'architecte. Là encore, la confusion est fréquente, probablement du fait que le rôle d'urbaniste est la plupart du temps le fait des architectes. Pourtant, l'urbaniste doit concevoir et faire évoluer le système d'information d'un point de vue global, tandis que l'architecte travaille dans un cadre local suivant des règles définies par l'urbaniste. Superviser le développement d'un quartier et construire des immeubles sont deux responsabilités différentes.

³ Edouard Jeanneret-Gris, dit Le Corbusier fut architecte, urbaniste et théoricien de l'urbanisme (1887-1965).

Chapitre III : Urbanisation des Systèmes d'information

Dans le cadre de l'évolution du système d'information, urbaniste et architecte sont des acteurs dont le rôle est prépondérant.

1.3.3 Les limites de l'urbanisme des SI

L'urbanisme des villes a des limites, celles-ci existent aussi pour les systèmes d'information. Ainsi, l'urbanisme (des cités) est-il impuissant à régler le problème de l'accroissement considérable de certaines mégalofoles, telle la ville de Mexico, capitale qui avale le Mexique, qualifiée de macrocéphale par les géographes. De la même façon, la ville peine à répondre aux demandes croissantes en matière de facilités de transport. Les polémiques autour du troisième aéroport parisien en sont une preuve.

Ainsi, c'est à la condition que la ville devienne l'affaire des citoyens eux-mêmes, qui sont aussi citoyens, que chaque habitant pourra se voir reconnaître un véritable droit de cité. De la même façon, les acteurs responsables des projets informatiques, maîtrise d'œuvre et maîtrise d'ouvrage doivent être sensibilisés à l'urbanisation. Les projets étant le moyen concret de faire évoluer le système d'information, il est impératif que ceux-ci soient réalisés en cohérence avec le plan d'urbanisme.

1.3.4 Démarches d'urbanisation et règles d'urbanisme

1.3.4.1 Présentations des démarches

Les principaux auteurs ayant publié des ouvrages sur l'urbanisation des systèmes d'information ont chacun une façon différente de concevoir ce concept et de ce fait, des approches différentes en termes de démarches. La principale distinction à faire concerne Jacques Sassoon. Précurseur du concept, il est important de noter que sa vision est orientée "urbanisation informatique" et que cette urbanisation se décline en un ensemble de règles. Ces règles, décrites en annexe, permettent de contrôler l'évolution du système informatique, dans le cadre d'un plan d'urbanisme. Il ne s'agit pas de réaliser un projet d'urbanisation à proprement parler, Jacques Sassoon insiste d'ailleurs bien sur ce fait :

L'urbanisme n'est pas une fin, mais un moyen. [...] L'urbanisme ne doit pas être considéré comme un projet en soi, mais comme un principe accompagnant chaque projet [Sassoon 98].

Ainsi, Sassoon ne préconise pas une démarche, mais plutôt un ensemble de règles à respecter pour mettre en place un système d'information urbanisé. Ceci diffère des démarches de Gérard

Jean et de Christophe Longépé qui proposent un ensemble d'étapes⁴ pour urbaniser le système d'information et qui voient plus l'action d'urbaniser comme un projet. Christophe Longépé a d'ailleurs intitulé son ouvrage : *le projet d'urbanisation du système d'information*.

⁴ En annexes 4 et 5, deux diagrammes synthétisent les étapes des démarches proposées par Gérard Jean et Christophe Longépé pour l'urbanisation des systèmes d'information.

Chapitre III : Urbanisation des Systèmes d'information

Cependant, quelle que soit la méthode, tous les auteurs s'accordent sur deux points, résultats directs de l'analogie avec l'urbanisme des cités, et préconisent la définition d'un équivalent du Plan d'Occupation des Sols (POS) accompagné d'un découpage du système d'information en sous-ensembles : zones, quartiers, et îlots indépendants.

1.3.4.2 Le Plan d'Occupation des Sols

Le Plan d'Occupation des Sols (POS) est un document d'urbanisme, fixant les règles générales d'utilisation du sol qui s'imposent à tous. Pour le système d'information, le POS fixe les règles d'utilisation des espaces du système d'information qui s'imposent à tous. En outre, il s'agit de définir le découpage en zones d'aménagement, et de spécifier les services et les responsabilités attachées à chaque sous ensemble. Ceci permet d'obtenir un système d'information organisé, pour lequel sont définis :

- l'objet et la mission des applicatifs le constituant,
- les regroupements d'applicatifs en ensemble cohérents,
- les périmètres réservés pour de futurs applicatifs (notamment transversaux).

Le POS doit prendre en compte les scénarii les plus probables d'évolution du système d'information.

Selon Christophe Longépé, les principaux éléments constitutifs du POS sont :

- un rapport de présentation dont l'objectif est la présentation des orientations structurantes pour le SI,
- un ensemble de cartographies présentant les sous-ensembles du SI,
- les règles d'urbanisme,
- la mission des différents sous-ensembles du SI.

1.3.4.3 La définition des sous-ensembles

Dans le paragraphe sur le POS, nous avons abordé le nécessaire découpage du système d'information en sous-ensembles cohérents. Comme pour une ville, il s'agit de définir des zones, quartiers, et îlots. Reprenons les définitions proposées par Jacques Sassoon pour ces notions.

Zone

Les traitements du SI sont découpés selon différents niveaux, correspondant aux préoccupations de temps et de métiers de l'entreprise. Ces niveaux sont dénommés zones [Sassoon 98].

Chapitre III : Urbanisation des Systèmes d'information

Pour le SI bancaire, Jacques Sassoon propose le découpage schématisé ci-après, auquel il préconise l'ajout d'une zone référentiel :

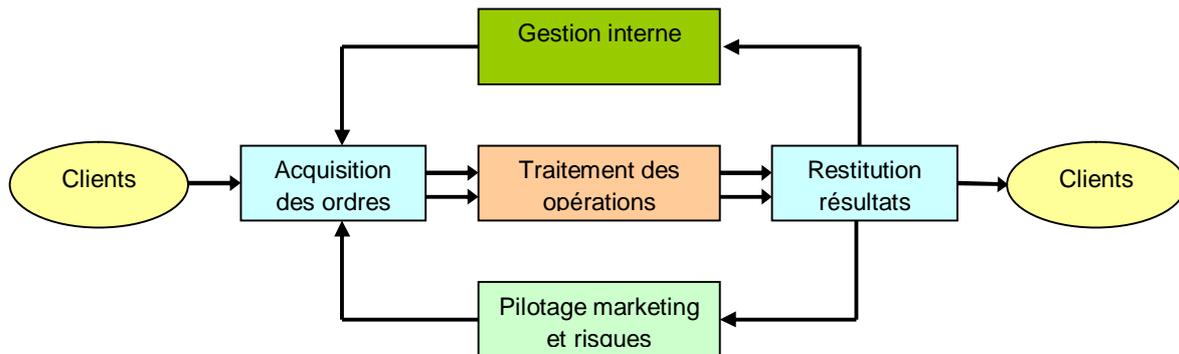


Figure 09 : Zones du système d'information dans les banques [Sassoon 98]

De la même façon, Christophe Longépé propose un découpage général du SI en zones. On retrouve ainsi :

- la zone échange, qui regroupe les zones acquisition et restitution du schéma ci-dessus,
- La zone décisionnel, équivalent de la zone pilotage de Jacques Sassoon,
- la zone opération,
- la zone ressource, qui correspond à la zone gestion interne,
- la zone référentielle regroupant les données communes à plusieurs applications,
- la zone gisement de données qui comprend toutes les informations dynamiques et pérennes de l'entreprise.

Quartier

Au sein de chaque zone, les traitements du SI seront répartis selon la nature des informations traitées. La nature de l'information définit un quartier. [Sassoon 98]

Le quartier correspond donc à ce que l'on appelle traditionnellement sous-système. Il a une certaine unité et une individualité.

Chapitre III : Urbanisation des Systèmes d'information

Ilot (ou bloc)

Un bloc est un ensemble de données et de traitements homogènes dans une activité de l'entreprise.⁵

Il s'agit de la plus petite unité de décomposition du système d'information. Le bloc recouvre une activité, il peut être composé d'un progiciel du marché.

Afin de représenter au mieux ces différents éléments, il est indispensable de disposer de modèles. En l'occurrence, on définira des cartographies du système d'information, identifiant le découpage du SI et les flux entre les différents sous-ensembles.

2.3.4.4 L'élaboration de règles d'urbanisme

L'élaboration de règles d'urbanisme permet de doter le système d'information d'un cadre général d'évolution. L'ensemble des règles proposées par Jacques Sassoon est repris en annexe 2 de ce document.

De la même façon, Christophe Longépé préconise un ensemble de règles composé de "règles d'urbanisme" et de "règles de bonnes pratiques".

Exemple de règle d'urbanisme pour l'architecture applicative, proposée par Christophe Longépé : " Le bloc offrant un service est le responsable de la qualité du service" [Longépé 01 (1)].

- Exemple de règle de bonne pratique pour l'architecture applicative, proposée par Christophe Longépé : " Toute architecture applicative comporte une zone de pilotage (ordonnancement) qui assure l'interface entre *front office*, *back office* et *middle office*".

3. Cartographie des systèmes d'information

Qu'il s'agisse de la démarche d'urbanisation proposée par Gérard Jean ou de celle proposée par Christophe Longépé (synthétisées sous forme de diagrammes en annexes 4 et 5), on trouve à chaque fois une étape consistant à modéliser l'existant et la cible du système d'information.

2.1 Modélisation du système d'information

La modélisation du système d'information est une technique de conception dans le cadre d'un projet d'évolution du système d'information. Le modèle est une abstraction du SI (un 'modèle

⁵ Ibid.

Chapitre III : Urbanisation des Systèmes d'information

réduit') plus facile à manipuler, ce qui permet de le visualiser et de le communiquer avant sa construction ou sa modification :

- ce n'est donc pas une simple documentation réalisée après la construction du SI,
- en tant qu'abstraction, un modèle est une simplification du SI qui ignore certains aspects jugés non essentiels et relégués au rang de détails,
- le modèle est donc spécialisé par la nature des aspects ignorés (c'est une certaine vue du SI),
- à partir d'une certaine complexité, il faut plusieurs vues pour construire un SI et chaque vue nécessite une notation spécifique.

Après différentes méthodes systémiques* dans les années 1970-80, la notation standard retenue depuis les années 1990 est celle de l'UML - Unified Modelling Language. Cependant, il est important de bien noter qu'UML n'est **pas une méthode** mais bien un langage de modélisation qui met à disposition une palette de diagrammes destinés à être utilisés pour la modélisation des applications informatiques. Il existe effectivement une méthode bâtie autour de UML, appelée UP (ou Unified Process) qui est fournie pour le processus de développement logiciel.

2.1.1 Pourquoi modéliser ?

Face aux exigences croissantes de compétitivité, de réactivité et d'efficacité, les entreprises ont besoin de systèmes d'information toujours plus performants. Elles sont de plus en plus nombreuses à souhaiter faire évoluer leur système d'information, notamment autour des technologies liées à Internet.

La technologie offre aujourd'hui une grande richesse de possibilités qui ouvre la voie à la construction des architectures de systèmes d'information les plus complexes et diversifiés. Les besoins toujours plus importants de communication entre applications requièrent une réflexion plus globale et plus approfondie de l'organisation des composants logiciels.

Par une représentation schématique et structurée, détaillée ou synthétique, la modélisation aide à concevoir et à décrire l'architecture des systèmes d'information, facilitant ainsi la maîtrise des évolutions ultérieures.

Il existe de multiples raisons de modéliser l'architecture d'un système d'information : l'architecte peut s'intéresser à la description de l'existant, en focalisant son analyse sur l'architecture logicielle et son intégration dans l'organisation, ou sur l'architecture technique et matérielle du système d'information ; il peut également étudier les évolutions nécessaires d'un système d'information, qu'il s'agisse d'intégrer une nouvelle application, un nouveau site Internet, ou un progiciel tel que **Erreur ! Source du renvoi introuvable.***, ou encore d'optimiser une architecture Client/Serveur, en recherchant la meilleure répartition des données et des traitements.

Chapitre III : Urbanisation des Systèmes d'information

L'architecte peut également être amené à représenter de manière précise les collaborations entre les applications de l'entreprise et celles de ses partenaires, ou entre des entités autonomes à l'intérieur d'une même entreprise comme lors de la mise en place d'un moteur de **Erreur ! Source du renvoi introuvable.***.

Par rapport à de simples représentations graphiques ou documents de présentation, la modélisation permet la constitution progressive d'une base d'information complète et parfaitement cohérente, facile à maintenir.

2.1.2 Que modéliser dans le cadre de l'urbanisation ?

Nous avons développé la nécessité de mettre en place des architectures cibles. Or il est indispensable de pouvoir communiquer sur ces architectures. Dans cette optique, il apparaît qu'une modélisation de ces architectures est nécessaire.

De même, il faut modéliser l'existant afin de savoir comment le faire évoluer au mieux, et afin de faire apparaître les incohérences et problèmes éventuels. En effet, nous avons vu les problématiques amenées par la croissance anarchique du SI. Très peu d'acteurs dans les entreprises possèdent une vision d'ensemble sur le système d'information et sont capables de détecter d'éventuelles redondances révélant des lacunes en termes de mutualisation. L'avantage de la modélisation est de permettre cette vision d'ensemble.

Il s'agit donc, dans le cadre d'une démarche d'urbanisation, de représenter les processus métiers supportés par le système d'information, l'organisation de ce système d'information en zones, quartiers et îlots, les applicatifs de ces sous-systèmes du SI, et enfin, les échanges de données ou flux d'information. Les cartographies permettent de fournir cette représentation.

2.1.3 Comment modéliser ?

Les outils de modélisation ont de plus en plus de succès et le langage de modélisation UML s'est progressivement imposé. Ce dernier met à disposition une palette de diagrammes (9 au total⁶) permettant de modéliser une application ou un ensemble d'applications. Dans le cadre de l'urbanisation des systèmes d'information, on cherche à représenter un ensemble important d'applications. Ce sont alors des cartographies qui sont utilisées pour représenter l'existant⁷. De fait, celles-ci sont au cœur de la démarche à suivre dans le cadre de l'urbanisation : une cartographie permet en effet d'appréhender la complexité et la dynamique du Système d'Information. Il s'agit de réaliser l'inventaire du patrimoine de l'entreprise après avoir

⁶ Un descriptif de ces diagrammes est fourni en annexe. □

Chapitre III : Urbanisation des Systèmes d'information

identifié les processus métier et analysé leur couverture fonctionnelle et applicative. Comme pour la cité⁸, la cartographie d'un SI est à la fois :

- Scientifique : elle repose sur un métamodèle.
- Artistique : il s'agit de communiquer, et, partant de là, l'esthétique est aussi un moyen de faciliter la communication.
- Technique : la réalisation s'appuie sur un certain nombre de techniques.

Les différentes cartographies utilisées dans le cadre de l'urbanisation des systèmes d'information, sont présentées dans la suite.

3.2 Cartographies et démarches

Définition :

La cartographie est définie comme l'ensemble des études et des opérations scientifiques, artistiques et techniques, intervenant à partir des résultats d'observations ou de l'exploitation d'une documentation, en vue de l'élaboration et de l'établissement de cartes, plans et autres modèles d'expression, ainsi que de leur utilisation. Définition proposée par Pierre Merlin et Françoise Choay dans le *Dictionnaire de l'urbanisme et de l'aménagement*, in [Longépé 01 (1)]

3.2.1 Différents types de cartographies

3.2.1.1 Cartographier l'architecture métier

La cartographie de l'architecture métier est en réalité une cartographie des processus. Elle permet :

- d'avoir une vue aérienne de l'ensemble des processus,
- de classer les processus en sous-catégories telles que :

§ processus opérationnels (ayant un impact direct sur la performance de l'entreprise ou de l'organisme),

§ processus de support (ayant un impact indirect sur la performance de l'entreprise ou de l'organisme).

⁸ Malgré les controverses sur le statut épistémologique de l'urbanisme évoquées par Pierre Merlin [Merlin 91], celui-ci s'accorde avec Françoise Choay dans l'ouvrage *Dictionnaire de l'urbanisme et de l'aménagement* pour voir en la cartographie un outil à base scientifique, artistique et technique.

Chapitre III : Urbanisation des Systèmes d'information

- d'identifier les liens qui peuvent exister entre plusieurs processus,
- d'établir une aide à l'ensemble de l'organisation à tous les niveaux de la hiérarchie de se "situer" dans une logique descriptive issue des processus,
- de faciliter en phase préalable la spécification et la mise en place d'un système d'amélioration de la performance.

Ce type de cartographie est décrit dans la figure .

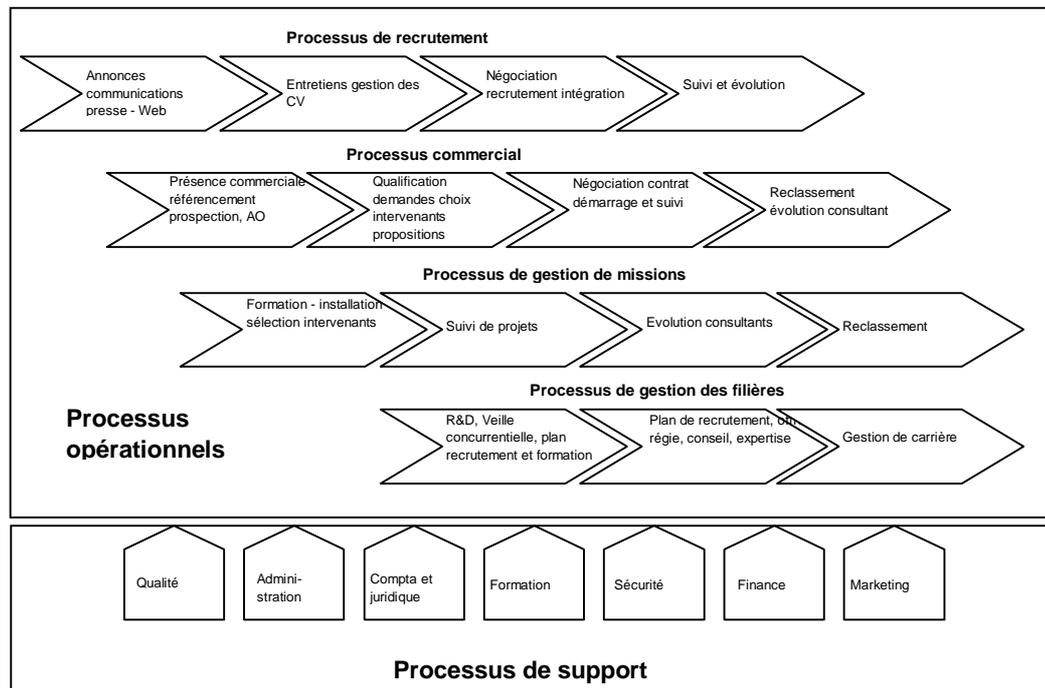


Figure 10 : Exemple de cartographie des processus : le cas d'une SSII [Longépe 01 (1)]

3.2.1.2 Cartographie l'architecture fonctionnelle

La cartographie fonctionnelle permet la description des fonctions du système d'information permettant de supporter les processus métier. Celle-ci se situe au niveau du système d'information.

Une telle cartographie permet de repérer les fonctionnalités similaires ou proches dans des applications distinctes et favorise la réutilisation.

1.2.1.2 Cartographie l'architecture applicative

La cartographie applicative décrit l'ensemble des éléments d'un système informatique. Ces éléments sont répartis en zones, quartiers et îlots, dans le but de limiter la portée des maintenances. Il faut donc réorganiser le système d'information, trouver les frontières entre

Chapitre III : Urbanisation des Systèmes d'information

les blocs, afin de le rendre modulaire et capable d'évolution. Cette démarche s'appuie sur deux idées clés, inspirées des concepts de l'orienté objet :

- cohérence forte / couplage faible : il s'agit de rendre les données et les traitements d'un même bloc cohérents (contraintes d'intégrité) et peu couplés aux données des autres blocs,
- **Erreur ! Source du renvoi introuvable.*** : un bloc est « propriétaire » de ses données et de ses traitements. Il propose des services aux autres blocs.

Une telle vision permet de limiter la portée des maintenances, rendre neutre vis-à-vis du SI, une modification dans le traitement d'un bloc, et enfin refondre le SI progressivement.

L'objectif d'une cartographie applicative est de montrer :

- les différents applicatifs composants le système informatique actuel. Sa granularité correspond à l'identification des zones, quartiers et îlots.
- Les flux entre ces applicatifs, afin de comprendre les dépendances entre les différents applicatifs.

La figure ci-après présente un exemple de cartographie applicative simplifiée.

Chapitre III : Urbanisation des Systèmes d'information

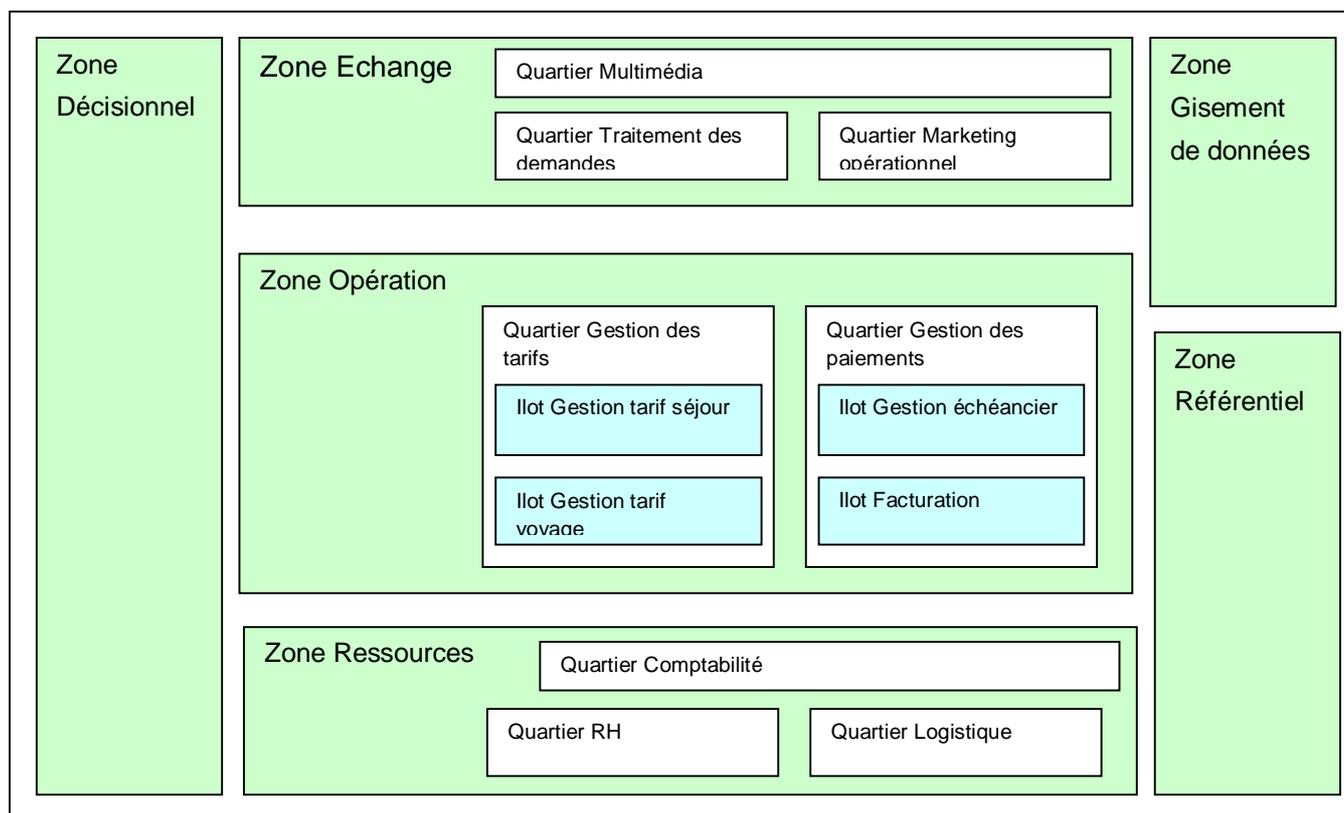


Figure 11 : Exemple simplifié de cartographie applicative : le cas d'un tour opérateur [Longépé 01 (2)]

La cartographie applicative est au cœur de la démarche d'urbanisation et se place au niveau du système informatique.

3.2.1.4 Cartographier l'architecture technique

La cartographie technique décrit les matériels, logiciels et les technologies utilisées.

3.2.2 Démarches

Nous avons vu que tout projet de réalisation d'une cartographie d'un système d'information s'inscrit dans un souci de pilotage de l'évolution du système d'information. Il s'agit d'avoir un état des lieux de l'existant et idéalement, un descriptif équivalent de la cible à atteindre. Dans toute entreprise, la documentation est souvent hétérogène en termes d'exhaustivité, de pertinence, de quantité... La démarche de cartographie vise à recenser l'existant en le modélisant de façon homogène.

Chapitre III : Urbanisation des Systèmes d'information

3.2.2.1 Première démarche : une action ponctuelle

Certaines sociétés choisissent de faire de la cartographie un projet global et mettent beaucoup de ressources sur un tel projet. Si l'on prend l'exemple de la Poste qui a réalisé une cartographie métier et applicative de son SI Production Courrier, 5 personnes en poste ont été affectées au projet, accompagnées d'un consultant à plein temps et d'un consultant à mi-temps. Le périmètre des applications à cartographier comptait une trentaine d'applications pour un projet d'une durée de 5 mois (3 mois pour la cartographie applicative, faisant apparaître les applications, les flux entre ces applications et les données gérées, et 2 mois pour la cartographie des processus et procédures).

Ce type de démarche présente l'avantage d'aboutir à une cartographie intégrale du SI en peu de temps. Cependant, la validité des informations recueillies dépend de l'implication des interlocuteurs fournissant ces informations. Or l'implication de ces interlocuteurs doit être motivée par leurs hiérarchies. En outre, la période pour réaliser la cartographie peut être importante : face à un problème de production, un responsable d'application ne verra pas l'urgence de la cartographie qu'il considèrera comme un travail documentaire.

3.2.2.2 Seconde démarche : une action récurrente

Il est également possible de traiter la cartographie comme une action récurrente et affiner au fur et à mesure les informations qu'elle contient. C'est la démarche choisie par Natexis Banques Populaires pour sa cartographie applicative. Une personne se trouve ainsi à plein temps sur la cartographie. Ainsi, après une première version qui s'apparente plus à un simple dictionnaire des applications, une deuxième version doit venir enrichir la première des flux entre les applications ainsi que des flux externes. Une troisième version devra permettre un recensement complet des fonctionnalités des applications, ainsi que des bases de données. Celle-ci devrait aboutir sur un lien avec une cartographie fonctionnelle.

L'avantage de cette démarche est la remise en cause continuelle des informations de la cartographie. Le système d'information évoluant par le biais des projets, il faut tenir compte de ces évolutions. Les versions successives sont un moyen de valider le contenu de la cartographie, tout en enrichissant celle-ci. L'inconvénient d'une telle démarche est que les informations contenues dans la cartographie sont incomplètes et qu'il faut communiquer sur ses évolutions si l'on souhaite que celle-ci devienne un réel outil de travail.

Chapitre III : Urbanisation des Systèmes d'information

3.2.2.3 Faire vivre la cartographie

En comparant les deux méthodes, il apparaît que l'avantage de la seconde sur la première est qu'elle permet de faire vivre la cartographie. Les systèmes d'informations évoluent, et s'il est important de cadrer leur évolution, il est également indispensable de pouvoir disposer d'une image fiable du système d'information. La cartographie, à l'image d'un site internet doit donc vivre s'il elle veut répondre à ses deux principaux objectifs :

- fournir une documentation fiable et complète sur le système d'information,
- être une documentation de référence.

Ainsi, il faut faire des mises à jour régulières de cette cartographie. Que celle-ci soit mise en place selon une démarche ponctuelle ou selon une action récurrente, il apparaît un moment où la cartographie est complète. A partir de ce moment-là, il est indispensable que chaque évolution du système d'information y soit consignée. Il faut donc inscrire dans les projets une étape incontournable de mise à jour de cette cartographie. Cela peut se faire en insérant cette étape dans le cycle de vie des projets. Par ailleurs, il est indispensable de nommer des responsables pour ces mises à jour.

3.3 Intérêt des cartographies pour le pilotage de l'évolution du SI

3.3.1 Positionnement des cartographies dans le cadre de l'urbanisation

Les cartographies sont des étapes indispensables de toute démarche d'urbanisation. On les voit apparaître aussi bien dans la démarche proposée par Gérard Jean que dans celle proposée par Christophe Longépé.

Si l'on considère les différents plans qui sont le métier, le SI et le système informatique, on peut dresser la matrice suivante, représentant le cadre général des travaux d'urbanisation et faisant apparaître notamment les positionnement des différentes architectures évoquées dans la partie 1 et modélisées par des cartographies.

Cette vue met en avant la nécessité de faire des zooms successifs jusqu'à la modélisation dans le cadre des projets qui sont les seuls vecteurs de l'évolution du système d'information.

Chapitre III : Urbanisation des Systèmes d'information

	Plan métier	Plan Système d'Information	Plan Système Informatique
Niveau Organisation	- Axes stratégiques (DG)	Architecture fonctionnelle urbanisée cible	- Architecture applicative - Architecture technique
Niveau Métier (domaine)	- Axes stratégiques du métier, - Schémas directeurs	Architecture fonctionnelle urbanisée cible du métier	- Architecture applicative du domaine métier
Niveau Projet	- Modélisation des processus métier du projet - Cahier des charges fonctionnel du besoin	Architecture fonctionnelle urbanisée cible du bloc fonctionnel traité par le projet	- Architecture des applications - Architecture technique des application

Figure 12 : Matrice du cadre général des travaux d'urbanisation

3.3.2 Avantages pour piloter l'évolution du système d'information

De même qu'un bâtiment ne saurait être construit sans un plan d'ensemble qui permet de comprendre la logique des connexions entre ses différentes parties, un SI ne saurait être élaboré sans un plan semblable à un schéma général. Ce plan général permet, en particulier, d'identifier toutes les conséquences d'une modification d'une partie de ce système.

La cartographie est l'outil de base indispensable à toute maintenance évolutive. [Mingasson 00]

Ainsi, la cartographie permet de représenter l'existant ainsi qu'une trajectoire intelligible pour les décideurs vers un système cible. Les cartographies modélisent les processus, les fonctions importantes, les applications et les interactions entre ces éléments. Elles permettent d'explicitier la structure des systèmes, d'évaluer la dynamique de leur évolution, de détecter les défauts structurels.

Elles permettent de prendre des décisions sur le système :

- le laisser évoluer de lui-même,

Chapitre III : Urbanisation des Systèmes d'information

- transformer sa structure par de grands travaux,
- baliser son évolution dans un cadre donné.

La plupart du temps, compte tenu de la sédimentation des systèmes d'information, il est nécessaire d'entamer une structuration volontaire et délibérée qui passe par de grands travaux de type "schémas directeurs" et par un accompagnement des projets qui sont les moteurs de l'évolution du système d'information. Par ailleurs, la transformation du système d'information doit le rendre flexible, adaptable, voire « adaptatif ». Il s'agit de le structurer de telle manière qu'il soit porteur de ses fonctions, mais également de ses potentialités. Il doit devenir apte à changer, à évoluer plus facilement en cas de besoin, à s'adapter avec un minimum d'efforts aux aléas imposés par l'environnement.

3.4 Contexte de la cartographie

3.4.1 Situation de départ

3.4.1.1 Motivation de la cartographie

Certaines applications effectuaient les mêmes traitements et il fallait déterminer la cible des applications à conserver et de celles à abandonner. Ainsi, la première demande de cartographie s'est faite dans le cadre d'un plan de convergence. Il est apparu nécessaire à la direction des Systèmes d'Information, de disposer d'un document de référence commun faisant émerger la cible et regroupant des informations sur les applications. La cartographie réalisée devait faire apparaître l'origine des applications et les applications à abandonner. Pour ce faire, l'existant à reprendre consistait en une cartographie très détaillée réalisée en 1998, et un document (des fiches Word) par application. Ces informations se plaçant à des niveaux différents, il fallait les harmoniser en choisissant le niveau de détail. La décision a été prise de se placer à un niveau très général et de proposer pour chaque application :

- une description fonctionnelle macroscopique,
- les informations de base sur les caractéristiques techniques.

Parallèlement, un premier travail d'urbanisation a été réalisé, en organisant l'existant par zones, quartiers, îlots et blocs.

Chapitre III : Urbanisation des Systèmes d'information

3.4.1.2 Première version de cartographie

A la fin de l'année 2000, deux documents Word ont été réalisés : un descriptif du système d'information et un dictionnaire des applications.

Le descriptif du SI avait pour objectif d'être une description simplifiée du système d'information et comportait les éléments suivants :

- Un tableau constituant la **représentation matricielle** du système :
 - les colonnes indiquant les différents métiers de la banque (crédits, marchés, moyens de paiement...),
 - les lignes indiquant les différentes organisations : Front Office (agence, salle des marchés...), Back Office, ainsi que les outils d'échanges, les outils de suivi et pilotage, et les référentiels,
 - chaque application du système d'information est affectée, d'une part à un (ou plusieurs) métier(s), d'autre part à une (ou plusieurs) organisation(s) : ceci définit sa (ou ses) place(s) dans la matrice.

		Assurance		Compte	Finances	Marchés	Moyens de paiements			Gestion interne	...
		Vie	Prévoyance	client			Chèques	Carte	...		
Front Office	Minitel										
	Internet										
	Agences										
	...										
Back Office	traitements										
Echanges	internes										
	place										
Référentiels											
...											

Figure 13 : Représentation matricielle des applications [Dub, 02]

- **Un schéma global d'urbanisation** de l'ensemble du système : ce schéma étant issu de la représentation matricielle et la simplifiant (de façon à rendre chaque application plus immédiatement accessible de façon visuelle).
- Un schéma présentant la **vision organisationnelle du système** : celui-ci reprend le diagramme d'urbanisation précédent, en affectant à chaque application une couleur différente selon son domaine gestionnaire.
- **Le schéma d'urbanisation y est ensuite décliné par métier**, chaque « vision par métier » représente les applications :
 - traitant les opérations du métier,

Chapitre III : Urbanisation des Systèmes d'information

- ou liées par échange direct de flux avec les applications traitant ces opérations.
- Un tableau récapitulant l'ensemble des applications par ordre alphabétique, indiquant pour chacune :
 - l'application et son libellé (décrivant succinctement ce que traite cette application),
 - sa position dans la matrice (organisation / métier),
 - l'origine de l'application.
 - la documentation disponible :
 - selon la cartographie intranet pour NBP,
 - selon les fiches par applications pour NB.
- le domaine gestionnaire de l'application.

Le dictionnaire des applications fournissait quant à lui, pour chaque application :

- ses fonctions,
- ses composants,
- ses caractéristiques (multi-établissement - plate-forme d'exécution - environnement de développement - moniteur - SGBD).

En 2001, ces informations ont été modélisées à l'aide du logiciel MEGA acquis dont une description est fournie en annexe 1.

3.4.1.3 Problématique

Il apparaissait nécessaire de mettre en place une nouvelle et unique version de la cartographie, modélisée sous MEGA et reprenant l'ensemble des informations disponibles à la fois sur la cartographie 1998 et la cartographie 2001 qui sera appelée dans la suite cartographie version 1.

L'objectif pour cette nouvelle version comprenait trois volets :

1. La mise à niveau des informations dans leur forme actuelle :

- prise en compte des évolutions (nouvelles applications, applications modifiées ou supprimées),
- introduction des applications censées disparaître en 2000 et qui sont toujours présentes dans le SI, validation des informations de la version 1.

Chapitre III : Urbanisation des Systèmes d'information

2. **La reprise des informations présentes dans l'ancienne cartographie (1997-1998) :** cette reprise se faisant au cas par cas, application par application (200 applications en commun entre les 2 cartographies). Dans le cas où les responsables de l'application considèrent que cette reprise est pertinente, une partie des informations de l'ancienne cartographie pourront être renseignées dans MEGA (notamment pour les fonctionnalités et les stockages décrits dans la cartographie 98).
3. **Le recensement des flux :** ce recensement devait se faire sur les flux externes et sur les flux inter-application. Il était motivé par un besoin en termes d'analyses d'impacts. Un recensement exhaustif des flux permettant en effet lors de l'évolution d'une application, de repérer l'intégralité des applications échangeant des informations avec celle-ci et susceptibles de subir un impact lors de cette évolution.

3.5 La cartographie s'inscrit dans le cadre de l'urbanisation

Nous avons présenté dans la partie 3.4.1.2 un tableau positionnant les différentes cartographies dans le cadre général d'une organisation. Cependant, il est évident que le passage d'une démarche d'urbanisation au niveau d'une entreprise à une démarche opérationnelle visant à urbaniser un système d'information ne peut que se faire dans les projets. Il faut donc sensibiliser les équipes à cette démarche. Le premier avantage de la cartographie est donc de fournir un premier document proposant une vision urbanisée du système d'information. Les équipes pourront donc s'appropriier les concepts et les différentes zones d'urbanisation du système d'information. Il est cependant indispensable d'organiser un accompagnement effectif des projets informatiques.

Ainsi, si on se contente de la vision d'urbaniste et qu'on ne descend pas au niveau des projets, le risque est que ce qui est réalisé dans le cadre de l'évolution du SI ne corresponde pas à ce qui se trouve sur les plans. Il faut savoir "décoder" les plans et les mettre en œuvre.

Dans le cadre de l'urbanisme on cherche essentiellement à :

- favoriser la réutilisation,
- découpler les systèmes pour les rendre plus évolutifs.

Il faut favoriser ceci au niveau des projets informatiques.

Chapitre III : Urbanisation des Systèmes d'information

Finalement, il apparaît que les différentes cartographies réalisées dans le cadre de l'urbanisation des systèmes d'information constituent un référentiel du système d'information qui est un actif précieux de l'entreprise ou l'organisme, permettant notamment :

- des analyses d'impact rapides et fiables,
- la maîtrise du SI,
- la formation rapide de nouveaux intervenants sur le système d'information.

Ces cartographies doivent faire l'objet de mise à jour régulières, sans quoi les informations de ce référentiel deviennent vite non fiables et même obsolètes.

Conclusion

Dans le contexte économique actuel, les entreprises doivent gérer leurs systèmes d'information avec rigueur et cohérence. Le système d'information est en effet au cœur du fonctionnement de toute organisation et son efficacité en conditionne les performances. La plupart du temps, les entreprises se trouvent confrontées à des systèmes d'information sédimentés. Ces derniers portant le poids d'une évolution dirigée par les contraintes du moment et les intégrations successives de nouvelles technologies.

L'urbanisation des systèmes d'information permet d'apporter de la cohérence dans le SI et d'en piloter l'évolution. La description du système d'information par les cartographies permet aux responsables des SI d'avoir un état des lieux de l'existant et de visualiser la cible vers laquelle les projets informatiques doivent permettre de tendre. Ainsi, les cartographies permettent d'explicitier la structure du SI, d'évaluer la dynamique de son évolution et de détecter les défauts structurels.

Sur la base de ces cartographies, de grands travaux peuvent être décidés pour structurer de façon volontaire et délibérée le SI et le rendre ainsi plus efficace, moins coûteux, plus homogène. Parallèlement, les évolutions classiques doivent être cadrées afin de rester dans la cible du SI urbanisé. L'objectif étant de parvenir à un système d'information flexible et aisément adaptable aux changements de technologies comme aux bouleversements structurels pouvant survenir dans les organisations.

L'urbanisation des systèmes d'information est donc une technique s'attachant à piloter l'évolution du système d'information. Les cartographies qui sont réalisées dans ce cadre peuvent donc devenir de véritables outils de travail pour les responsables des projets informatique.

Glossaire

Application informatique

Une application informatique est un ensemble de programmes articulés entre eux, utilisés pour automatiser ou assister des tâches de traitement de l'information dans un domaine particulier.

Cas d'utilisation fonctionnel (ou CUF)

Le cas d'utilisation fonctionnel représente une façon d'utiliser le système d'informations au bénéfice d'un ou plusieurs acteurs.

Encapsulation

Principe de la programmation orientée objet consistant à cacher les données et les algorithmes des objets en ne laissant visibles que les interfaces. La notion d'encapsulation signifie que les données, de même que les méthodes permettant de les manipuler, sont contenues dans un même emballage (objet) et qu'un utilisateur (humain ou un autre objet) ne peut directement atteindre les données associées à cet objet. Ce dernier sera donc uniquement concerné par la requête de services à effectuer et non par les détails et la façon dont ces services sont effectués.

Information

Ensemble des faits et connaissances déduits des données. Un ordinateur manipule et génère des données. La signification déduite des données est l'information : c'est-à-dire que l'information est une conséquence des données. Les deux mots ne sont pas synonymes bien qu'ils soient souvent utilisés l'un pour l'autre.

Méthode systémique

Les méthodes d'analyse systémique, appartiennent au courant scientifique qui analyse les éléments de processus complexes comme composants d'un ensemble où ils sont en relation de dépendance réciproque. Leur champ d'étude ne se limite pas à la mécanisation de la pensée : l'analyse systémique est une méthodologie qui organise les connaissances pour optimiser une action.

[Tapez le titre du document]

En informatique, les méthodes systémiques apparues vers les années 75-80 appréhendent de manière globale le SI. L'approche est conceptuelle et distingue quatre niveaux d'abstraction :

- le **niveau conceptuel** exprime les choix de gestion ; il consiste à se poser la question "quoi faire" ?
- le **niveau organisationnel** exprime les choix d'organisation des ressources humaines et matérielles, au travers notamment de la définition des sites, des postes de travail. On intègre à ce niveau les notions de lieux, de temps et d'acteurs : on doit se poser les questions suivantes : qui, où, quand, et combien ?

Avec les deux autres niveaux, on franchit la frontière des systèmes d'information informatisés et l'on va se poser des questions de moyens informatiques : comment ?

- le **niveau logique** exprime les choix de moyens et ressources informatiques en faisant abstraction de leurs caractéristiques techniques précises.
- Le **niveau physique** traduit les choix techniques et prend en compte leur spécificité.

Procédure fonctionnelle

Une procédure fonctionnelle est une partie de la dynamique du système d'information qui assure le traitement d'un événement ou la production d'un résultat, de façon organisée dans le temps et dans l'espace.

Processus

Succession d'étapes élémentaires permettant d'aboutir à un résultat (objectif du processus). Les processus permettent de décrire les systèmes. Pour une entreprise, un processus est une partie de la dynamique d'un métier en tant que système. Ainsi, on peut définir le processus métier comme "l'enchaînement des activités à valeur ajoutée, qui aboutit à délivrer un produit ou service à un segment de clientèle. Il se décompose en processus opérationnel, de pilotage et de support. " [Jean 00]

Progiciel

Un progiciel est un ensemble complet de documents et de programmes conçu pour être fourni à plusieurs utilisateurs, en vue d'une même application ou d'une même fonction à partir d'un même type de données (produit + logiciel). Un progiciel simplifie en principe un ensemble de tâches qu'il permet d'organiser et auxquelles il offre une interface graphique ergonomique d'utilisation.

SAP

SAP est un progiciel de gestion intégré (ERP ou Enterprise Resource Planning) Il s'agit d'un ensemble de solutions de gestion décomposé en modules couvrant l'ensemble des activités d'une entreprise (ressources humaines, gestion de la chaîne logistique, gestion de la relation client, cycle de vie produit)

UML (Unified Modeling Language)

L'UML est le concept de modélisation objet. Son développement a débuté en 1994 et il a été normalisé par l'OMG début 1997. Grady Booch et Jim Rumbaugh ont décidé de travailler ensemble pour réaliser une unification des anciennes méthodes à objet. La méthode UML est très intuitive, plus simple et plus cohérente que les autres méthodes. Elle simplifie le processus complexe de conception d'un système informatique en définissant 9 points de vue (ou diagrammes) de modélisation.

UP

UP ou Unified Process (Processus Unifié) est un processus de développement de systèmes d'information et d'applications logicielles couvrant l'intégralité du cycle de vie développé par la société Rational Software. Il regroupe les pratiques modernes de développement logiciels et propose aux équipes des guides, des recommandations et des modèles pour l'ensemble de leurs activités.

Les caractéristiques essentielles du processus unifié, sont au nombre de cinq :

- RUP est à base de composants,
- RUP utilise le langage UML,
- RUP est piloté par les cas d'utilisation,
- RUP est centré sur l'architecture,
- RUP est itératif et incrémental.

Le pilotage par les cas d'utilisation fait de UP un processus de développement centré sur l'utilisateur (humain ou représenté par un autre système). Il permet une meilleure définition des besoins fonctionnels et une référence régulière à ces spécifications.

Workflow

Le workflow fait partie des moyens informatisés nécessaires à la communication entre acteurs d'une organisation, à leur coopération et à la coordination de leurs actions. Ces moyens sont regroupés sous la dénomination de groupware. Le Workflow permet l'automatisation de la circulation des documents nécessaires au travail entre individus d'une organisation voire au delà de cette organisation. Le terme workflow se traduit par "gestion du flux de travail"

Bibliographie

Ouvrages, Mémoires et thèses

[Avignon Joguet Pezziardi 00] Avignon L., Joguet D., Pezziardi P. – *Intégrations d'applications, l'EAI au cœur du e-business*, Edition Eyrolles, Paris, 1^{ère} édition, 2000.

[Biga, 2006] : Michel Bigand. *Conception des systèmes d'information: modélisation des données, études de cas*. Editions TECHNIP, 2006

[Bouchy, 1994] : S.Bouchy. « *L'ingénierie des systèmes d'information évolutifs*». Eyrolles, 1994.

[Brau,1995] : Sébastien BRAU, Christian CHANE-NAM, Louis-Laurent ANTIGNY, Gilberto DE VASCONCELOS, *L'ARCHITECTURE CLIENT-SERVEUR*, Octobre 1995.

[Canon 00] Canon F. – *Réalisation de connexions entre un référentiel de processus et une cartographie applicative*, Thèse professionnelle Mastère MSIT, 2000.

[CASTELLANI, 1987] : X.CASTELLANI. *macro-analyse étude préalable et analyse conceptuel des systèmes d'information*. Edition Masson, 1987

[Chéry 00] Chery S. – *Urbanisation fonctionnelle et intégration d'application*, Thèse professionnelle Mastère MSIT, 2000.

[Darras, 2004] : Darras.Franck .*proposition d'un cadre de référence pour la conception et l'exploitation d'un progiciel de gestion intégré*. Institut National Polytechnique de Toulouse.2004

[Dub, 99] : Jean-Charles Dubois, 'Un panorama des mille et un risques associés au bogue de l'an 2000', fevrier 1999.

[Dub, 02] : Stéphanie Dubois, 'La management de l'évolution du système d'information : Apports de l'urbanisme et des cartographies pour piloter l'évolution d'un système d'information ', octobre 2002.

[Falt, 2009] : Boi Faltings, Michael Ignaz Schumacher, 'L'intelligence artificielle par la pratique', ,2009.

[Gabay Gébré 99] Gabay J., Gebre B. – *La conduite des projets d'évolution dessystèmes d'information*, Dunod, Paris, 1999.

[Tapez le titre du document]

[Gaye 00] Gaye D. – *Architecture et évolution du système d'information : le cas de la banque de détail, Thèse professionnelle Mastère MSIT, 2000.*

[Inmon, 1996] : William H. Inmon, 'Construction de l'entrepôt de données', édition Broché, 1996.

[Jean 00] Jean G. – *Urbanisation du business et des SI, Hermès, Paris, 1^{ère} édition, 2000.*

[Lefeb, 2001] : Alain Lefebvre. *Le troisième tournant. Comment l'informatique professionnelle a évolué depuis ses débuts jusqu'à Internet et quels sont les développements inattendus qui vont suivre... Editions Dunod, Paris, 2001.*

[Le Moigne, 1999] : Le Moigne J.-L., *La modélisation des systèmes complexes, Dunod, 1999.*

[Leva, 2002]: Véronique Levasseur, *APPROCHE METHODOLOGIQUE POUR L'URBANISME DES SYSTEMES D'INFORMATION, juin 2002.*

[Levasseur 99] Levasseur V. – *Approche méthodologique pour l'urbanisation des systèmes d'information, Mémoire de DESS systèmes d'information, IAE, 1999.*

[Longépé ,01 (1)] Longepe C. – *Le projet d'urbanisation du système d'information, Dunod, Paris, 1^{ère} édition, 2001.*

[Mat, 1995] : J.P. Matheron. *Comprendre Merise : outils conceptuels et organisationnels. Editions BERTI. 1995*

[Merlin 91] Merlin P. – *L'urbanisme, Presses Universitaires de France, Col. Que Sais-Je, Paris, 1^{ère} édition, 1991.*

[Mingasson 00] Mingasson M. – *Informatique et stratégie d'entreprise : architecture et pilotage des SI, Dunod, Paris, 1^{ère} édition, 2000.*

[morley et al. 2005] : Morley, Chantal, Jeans Huges, Bernard Leblanc, et Olivier Huges. *Processus métier et système d'information : Evaluation, modélisation et mise en œuvre. Dunod 24 mars 2005*

[O'Bri, 1997] : James O'Brien, Guy Marion. *Les systèmes d'information de gestion. Editions Boeck Supérieur, 1997*

[Poul, 2011] : Alain Lefebvre & Laurent Poulain , *Cow-boys contre chemin de fer ou que savez-vous vraiment de l'histoire de l'informatique?*, Edition Talking Heads, 2011

[Reix 00] Reix R. – *Systèmes d'information et management des organisations, Vuibert, Paris, 3^{ème} édition, 2000.*

[Reix, 2004] : R.Reix. *Systèmes d'information et management des organisations. Editions Vuibert. 2004.*

[Tapez le titre du document]

[Rodh, 2010] : Florence Rodhain, Sylvie Desq, Bernard Fallery, Aurélie Girard., *Une histoire de la recherche en Systèmes d'Information à travers 30 trente ans de publications, article, 2010.*

[Rolland, 1990] : C. ROLLAND, C.PROIX. « vers une automatisation des processus de conception par les outils ».1990

[Sassoon 98] Sassoon J. – *Urbanisation des systèmes d'information, Hermès, Paris, 1998.*

[Soln, 96/97] : Christine Solnon, ' Ateliers de Génie Logiciel ', 1996/97

[TOUZI, 2007] : Jihed TOUZI. *Aide à la conception de Système d'Information Collaboratif support de l'interopérabilité des entreprises. THESE Pour obtenir le grade de Docteur de L'INPT, 2007*

[Vidal, 2009] : Vidal P., Augier M., Lacroux F., Planeix P. & Lecoer A., *Systèmes d'information organisationnels, Pearson Education (2 édition), 2009.*

[Volle 00] Volle M. - *Le pilotage du système d'information, Séminaire 2000 du club des Maîtres d'Ouvrage des systèmes d'information, 25-26 mai 2000*

[Tapez le titre du document]

Articles

[Bourdariat] Bourdariat J. – *Pourquoi faut-il renouveler le management des technologies de l'information et de la communication dans l'entreprise ?*

<http://www.itgi.org/itgovarticle.doc>

[Cotin 99] Cotin A. – *L'urbanisation sera un sujet majeur dans les cinq ans à venir, Le monde informatique, 1999*

[Martin 99] D. Martin, *Construisons l'architecture applicative avancée, article de l'informatique Professionnelle, n°172, 1999 in [Gaye 00]*

[Rémy 01] Remy C. – *L'urbanisation des systèmes d'information séduit de plus en plus les banques, 01 informatique, 2001*

Sites Internet

Site de Christian Braesch enseignant-chercheur à l'Université de Savoie

[1] : <http://www.braesch.fr>

Site de jean Bellec, gestionnaire pour le compte de la FEB de listes de diffusion sur l'histoire de l'informatique

[3] : <http://www.kerleo.net/>

Une base de données exclusive de dissertations gratuites pour les étudiants.

[4] : <http://www.dissertationsgratuites.com>

Site de l'information professionnel pour les décideurs

[5] : <http://www.guideinformatique.com>

[Tapez le titre du document]

Site de la communauté informatique

[6] : <http://www.commentcamarche.net>

Site du club des urbanistes et architectes des systèmes d'information

http://www.urba-si.com/index_public.php

www.oboulo.com

Site du club des maîtres d'ouvrage des systèmes d'information

<http://www.clubmoa.asso.fr/accueil.htm>

Site de Michel Volle, président du club des maîtres d'ouvrage des systèmes d'information

<http://www.volle.com>

[Tapez le titre du document]

Annexes

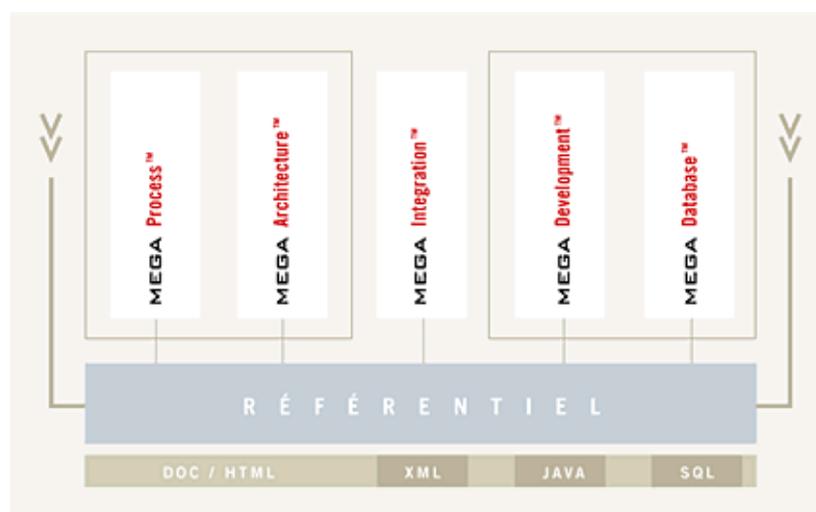
Annexe 01 : Présentation du logiciel MEGA [Dub, 02]

Depuis plus de 10 ans, MEGA développe une famille d'outils de modélisation permettant de répondre aux besoins d'un grand nombre d'entreprises. Les produits MEGA sont destinés aux équipes d'analystes, architectes et équipes de développement qui souhaitent améliorer leur efficacité en liant étroitement les évolutions rapides des processus et les systèmes qui les supportent. Grâce à son architecture ouverte basée sur un référentiel, la Suite logicielle MEGA permet un véritable travail d'équipe et une totale personnalisation à votre environnement.

MEGA Process, MEGA Architecture, MEGA Development, MEGA Database et MEGA Integration sont les outils majeurs d'analyse et de conception basés sur le référentiel MEGA.

Le référentiel MEGA est le cœur de la suite logicielle MEGA. Il fournit des services de stockage, d'administration, de reporting et de sécurité. MEGA Process permet d'identifier, définir et documenter les processus et les procédures de l'entreprise. MEGA Architecture assure la cartographie des applications et permet de construire une architecture de référence. MEGA Development et MEGA Database assurent la modélisation UML des composants applicatifs et des données, et la génération automatique de code, de scripts de bases de données et de schémas XML pour de nombreux environnements. MEGA Integration fournit des outils de spécification et de modélisation dédiés à l'intégration d'applications.

Grâce au référentiel MEGA, tous les produits automatisent la production de documentation et la publication de sites internet pour partager l'information avec toutes les équipes concernées, tant internes qu'externes.



Méga architecture

Avec MEGA Architecture, la Direction des Systèmes d'Information dispose à tout moment d'une vision globale des systèmes et des applications afin de mieux comprendre et décider, en améliorant qualité et rapidité de leurs évolutions.

MEGA Architecture permet aux concepteurs et architectes informatiques d'analyser et de modéliser les systèmes d'information et leurs relations avec les processus de l'entreprise. MEGA Architecture permet de définir une architecture de référence en mettant en relation les systèmes d'information et les flux de données et messages échangés.

MEGA permet de planifier l'évolution des systèmes d'information, faciliter l'intégration de nouvelles applications et l'amélioration des composants existants.

MEGA Architecture permet de décrire des systèmes d'information complexes et leurs liens avec des systèmes externes.

Grâce au référentiel MEGA, MEGA Architecture automatise la production de documentation et de sites web ; ceci assure le partage de l'information sans coût supplémentaire, au sein des équipes de projet et permet également une meilleure communication avec les prestataires externes.

Utilisé avec MEGA Process, MEGA Architecture reconstruit une vue cohérente des environnements fonctionnels et technique de l'entreprise, et permet de comprendre aisément les impacts des changements sur ces structures. Le résultat du travail des architectes est mémorisé dans le référentiel MEGA qui fournit une vue commune pour le travail d'équipe et la construction de solutions nouvelles. Cette source unique et cohérente garantit que prestataires et consultants suivent les normes et standard de l'entreprise. MEGA Architecture permet de piloter l'évolution de l'architecture du système d'information.

Annexe 02 : Les 7 règles d'urbanisme de Jacques Sassoon

1. Règle numéro 1 : Appartenance

Un bloc ne peut appartenir qu'à un seul quartier et à une seule zone. D'où un quartier ne peut appartenir qu'à une seule zone. [Sassoon 98]

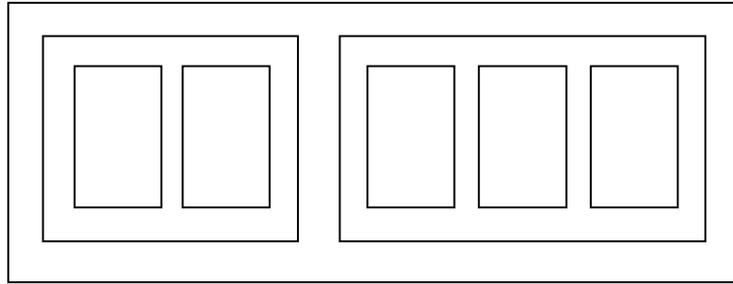


Figure 14 : Illustration de la règle d'appartenance [Sassoon 98]

2. Règle numéro 2 : Autonomie

Un bloc est indépendant des autres blocs s'il peut traiter de bout en bout un événement sans faire appel à un autre bloc. Une fois initié, un traitement n'est pas lié à l'attente de résultats émis par un autre bloc.

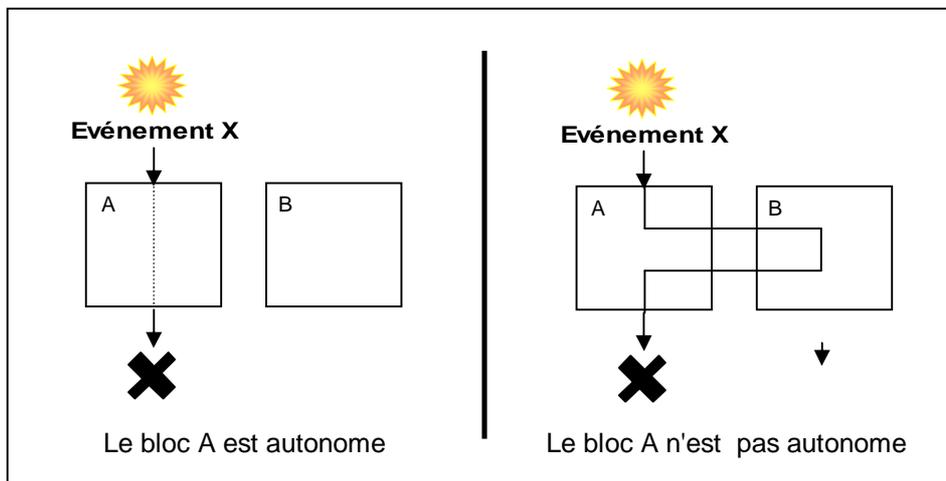


Figure 15 - Illustration de la règle d'autonomie [Sassoon 98]

3. Règle numéro 3 : Asynchronisme

Après avoir traité un événement, un bloc peut en traiter immédiatement un autre sans attendre le résultat d'autres traitements dans un autre bloc. [Sassoon 98]

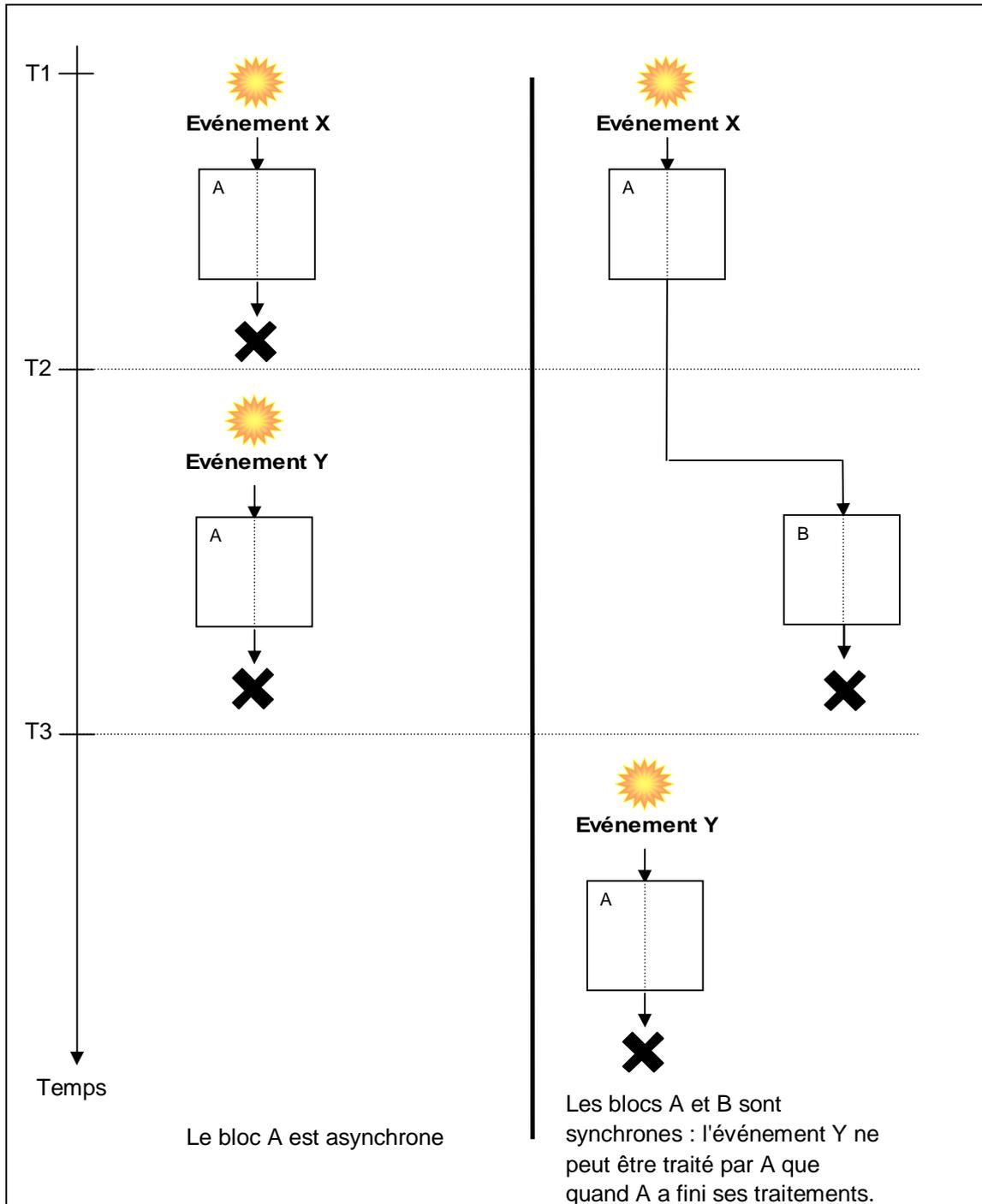


Figure 16 - Illustration de la règle d'asynchronisme [Sassoon 98]

4. Règle numéro 4 : Point d'ancrage

Un bloc traite un événement de bout en bout : un point d'ancrage "haut" (arrivée de l'événement à traiter) et un point d'ancrage "bas" (sortie du compte-rendu d'événement). Si les règles d'autonomie et d'asynchronisme sont respectées, le bloc n'a besoin que de deux points d'ancrage. [Sassoon 98]

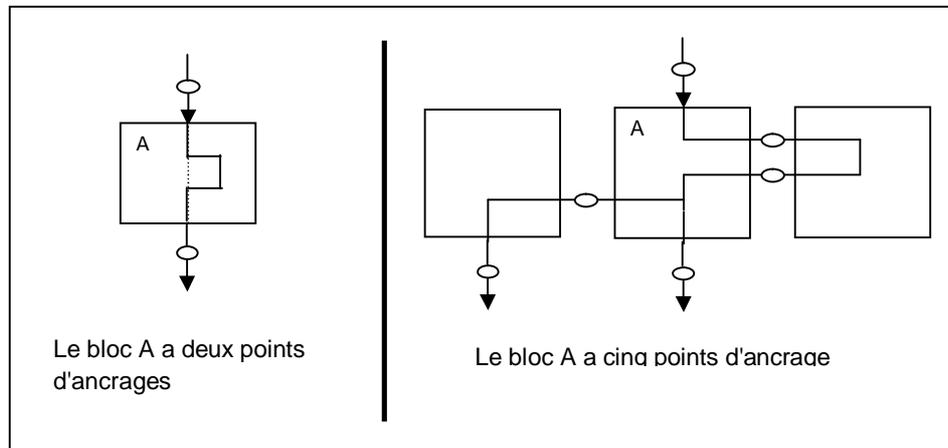


Figure 17 - Illustration de la règle des points d'ancrage [Sassoon 98]

5 .Règle numéro 5 : Propriété des données

Un bloc correspond à une activité et donc à des traitements et des données qui lui sont propres. Tout bloc peut avoir accès en consultation aux données d'un autre bloc (déclenchement de requêtes).¹

6 .Règle numéro 6 : Normalisation des échanges

Un flux représente un échange entre les différents acteurs du SI. Ces acteurs peuvent être les différents métiers et activités de l'entreprise (flux internes), mais aussi des acteurs extérieurs à l'entreprise (flux externe).

Il est important de normaliser les formats des flux entre les différents blocs (pour les échanges externes, il existe souvent une norme (ex : Swift, Edifact...)).

7. Règle numéro 7 : Gestion des flux

Toute communication entre les blocs transite nécessairement par le système de gestion des flux.

¹ Ibid.

[Tapez le titre du document]

Le système de gestion des flux permet, par l'enrichissement des interfaces et par leur normalisation, une réduction des adhérences ou imbrications entre les applications tout en simplifiant celles-ci.

Il doit assurer les 5 fonctions suivantes :

- l'administration des échanges (identification de l'émetteur, contrôle),
- l'interprétation des informations (enrichissement, transformations),
- l'agrégation de l'information,
- le routage et la gestion des files d'attente,
- l'ordonnancement des traitements.

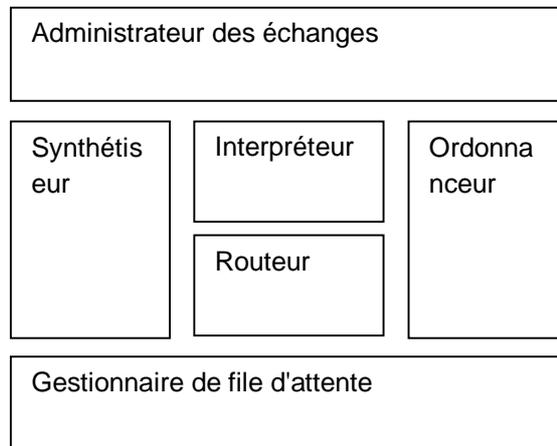


Figure 18 - Le gestionnaire de flux [Sassoon 98]

Le gestionnaire de flux est responsable de tous les flux entre différentes zones du SI comme l'illustre le schéma ci-dessous.

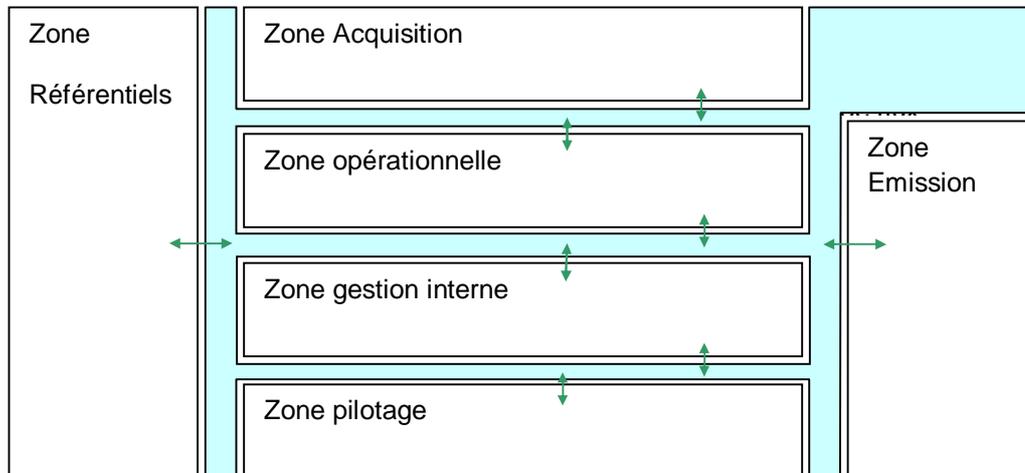


Figure 15 - Plan d'ensemble du SI et situation du gestionnaire de flux [Sassoon 98]

Annexe 03 : Les diagrammes d'UML

Il est important de bien noter qu'UML n'est **pas une méthode** mais bien un langage de modélisation qui met à disposition une palette de diagrammes destinés à être utilisés pour la modélisation des applications informatiques. Il existe effectivement une méthode bâtie autour de UML, appelée UP (ou Unified Process) qui est fournie pour le processus de développement logiciel.

Les diagrammes d'activité :

Représentation du comportement d'une opération en terme d'actions.

Les diagrammes de cas d'utilisation :

Représentation des fonctions du système du point de vue de l'utilisateur.

Les diagrammes de classes :

Représentation de la structure statique en termes de classes et de relations.

Les diagrammes de collaboration :

[Tapez le titre du document]

Représentation spatiale des objets, des liens et des interactions.

Les diagrammes de déploiement :

Représentation du déploiement des composants sur les dispositifs matériels.

Les diagrammes d'états-transitions :

Représentation du comportement d'une classe en terme d'états.

Les diagrammes d'objet :

Représentation des objets et de leurs relations, correspond à un diagramme de collaboration simplifié, sans représentation des envois de message.

Les diagrammes de séquence :

Représentation temporelle des objets et de leurs interactions.

Annexe 04 : La démarche d'urbanisation de Gérard Jean

Gérard Jean propose une démarche en 6 étapes, synthétisées dans le schéma suivant.

[Tapez le titre du document]

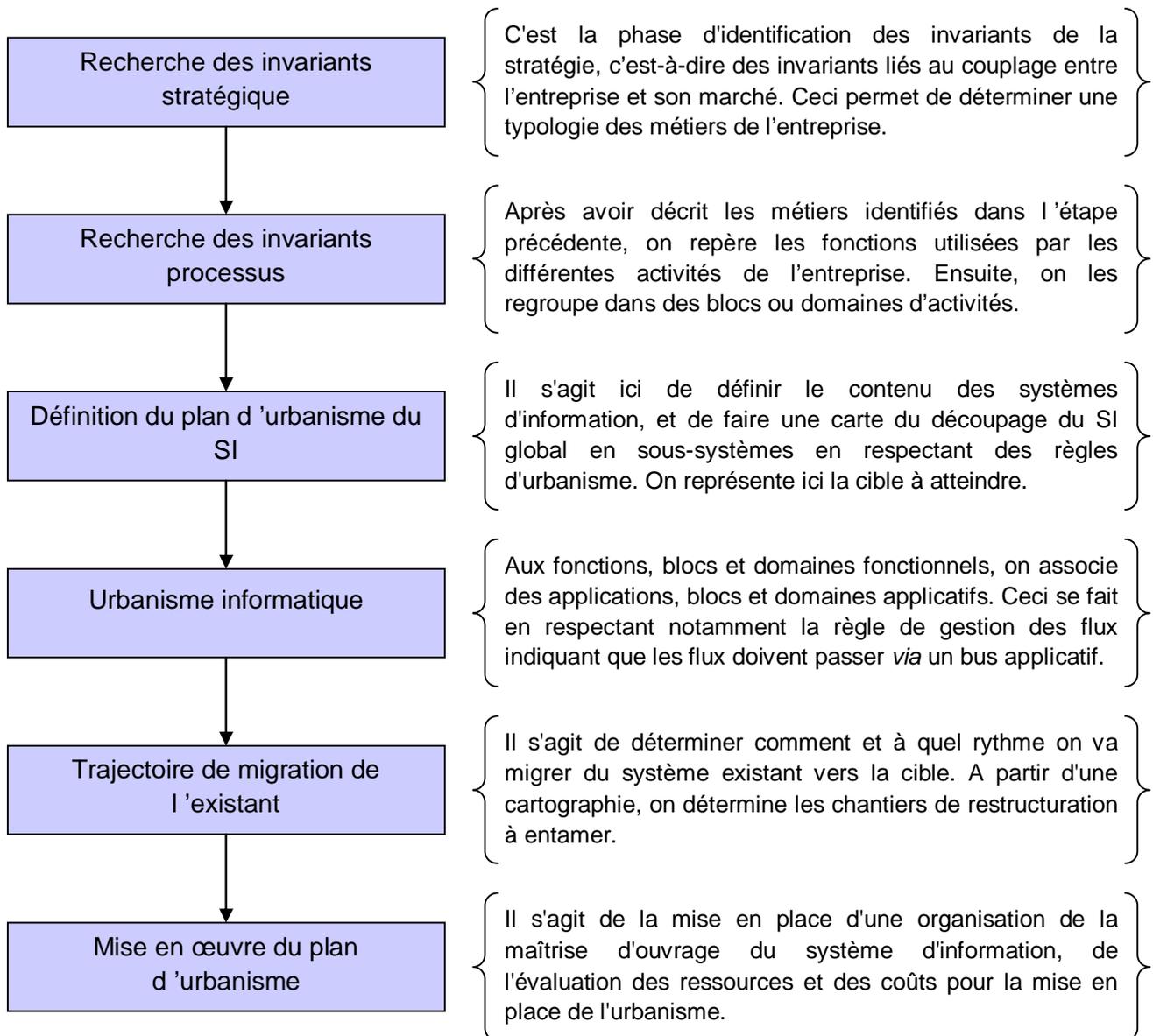


Figure 19 - Démarche d'urbanisation de Gérard Jean

Annexe 05 : La démarche d'urbanisation de Christophe Longépé

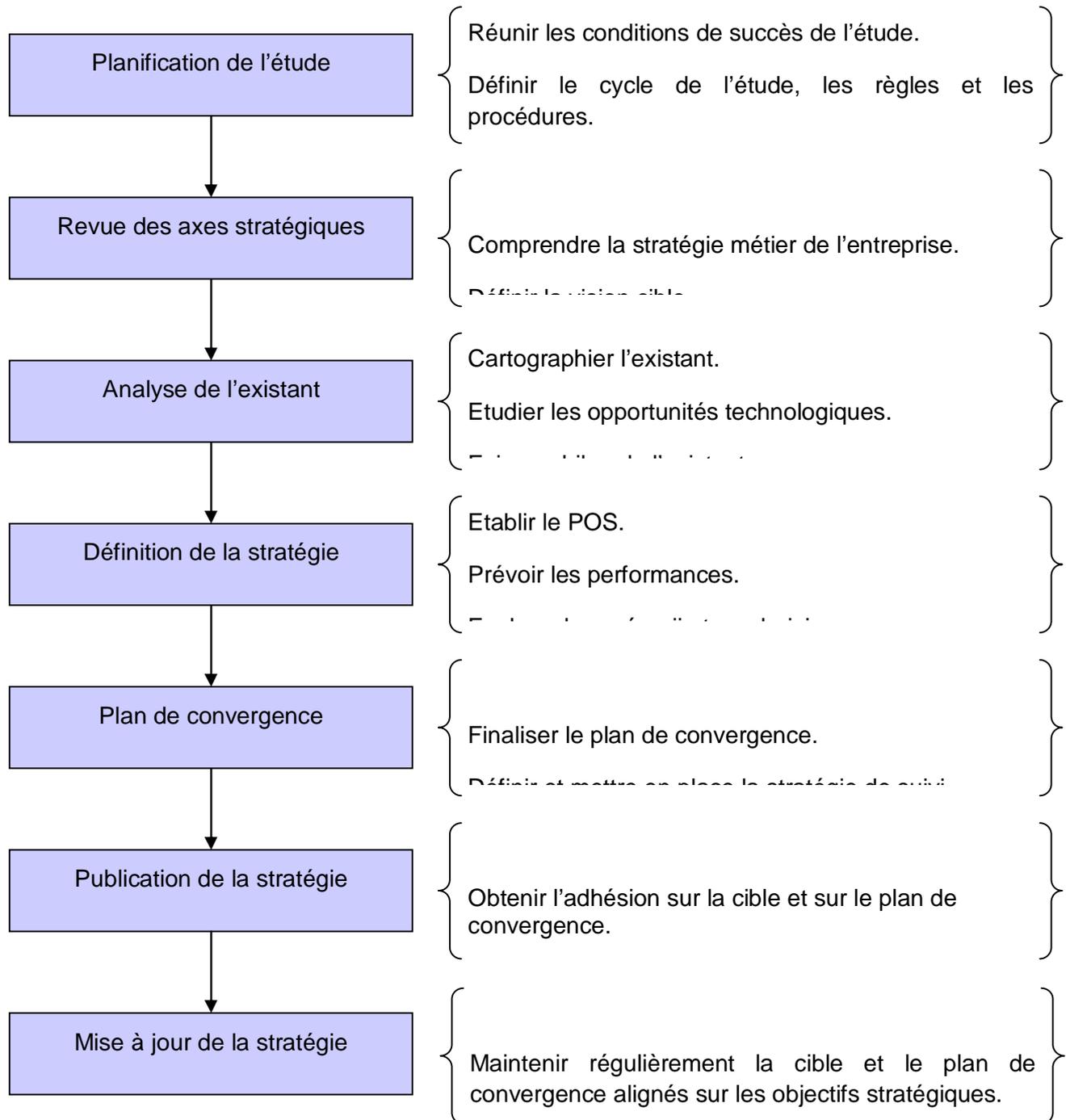


Figure 20 - Démarche d'urbanisation de Christophe Longépé

Conclusion générale

Conclusion Générale :

La recherche de l'urbanisme ou urbanisation du système d'information est une tâche particulièrement ardue, quand on veut la mener a posteriori

Si on veut éviter de procéder à des rafistolages, il est clair qu'il faut commencer par avoir une politique d'entreprise, que cette politique doit être adaptée aux possibilités techniques du moment (ou à celles attendues dans un avenir proche).

A partir de là, une stratégie d'informatisation doit être définie, stratégie qui se mettra au service des utilisateurs potentiels, selon les plans de la direction.

Et cette stratégie doit être à la portée des maîtres d'œuvre qui auront à se l'approprier pour la faire appliquer, en utilisant les démarches, normes et outils structurés les mieux adaptés.

Alors, si tout se déroule conformément aux plans, le système d'information disposera d'une cartographie lisible, correspondant à la réalité, son plan d'urbanisme.

Un système d'information urbanisé c'est finalement remettre l'informatique au service du métier, établir la relation entre les systèmes informatiques et la stratégie de l'entreprise, organiser les briques du système d'information dans un objectif de souplesse, et décliner puis intégrer progressivement les demandes d'évolution du système d'information par une approche rationalisée.

Car la finalité de la démarche d'urbanisation c'est la vie du SI, son évolution, sa flexibilité et sa pérennité et non pas la seule connaissance photographique à un instant t, qui n'est qu'une étape. L'urbanisme n'a de valeur ajoutée pour l'entreprise qu'en tant que garant de la réactivité continue de son SI.

Nous croyons que ce présent travail a illustré l'intérêt que d'urbaniser un système d'information en suivant les différentes étapes appropriées. Cette nouvelle discipline porteuse d'espoirs mais encore très jeune. Même si quelques pionniers chercheurs, consultants ou experts disposent déjà de retours d'expérience significatifs et de méthodologies formalisées, elle connaît tout de même certaines limites qui tracent à n'en pas douter les perspectives de recherche pour les années à venir.

Nous concluons donc notre travail par quelques perspectives qui consiste en :

- La nécessité de réinventer les relations MOA (maîtrise d'ouvrage)/MOE (maîtrise d'œuvre)
- Les limites des processus de constitution du plan d'urbanisme
- Les limites liées à la stabilité et la maturité des technologies
- limites liées aux performances.
- limites des processus de mise en œuvre du programme d'évolution.