

1

Étape 1 - La préparation du projet

Comme nous l'avons souligné en introduction, XML permet d'augmenter la flexibilité des systèmes d'information. Toutefois, un tel résultat ne s'obtient pas sans plan de travail : une méthodologie adaptée est nécessaire. Commençons donc par le début : la réalisation d'une esquisse du futur système.

Ce chapitre est consacré à ce que vous devez faire dans les tout premiers temps de votre projet. On pourrait appeler cette période le projet de projet.

Nous y expliquons comment réaliser rapidement un premier dégrossissage du problème qui se présente à vous. Cela doit vous permettre non seulement de comprendre le projet, mais encore de le partager avec d'autres alors que seules des idées sont formulées (à ce stade, rien de concret n'existe encore vraiment).

Nous présentons ici la méthode qui permet d'obtenir rapidement l'esquisse du projet en tenant compte, dès le départ, de grands principes que nous expliquons au cours de ce chapitre.

Inspirés des méthodes de conception d'*architectures orientées services*, les croquis (ou esquisses) que vous obtiendrez représenteront *in fine* les fonctions, sous-fonctions et services qui composeront le futur système.

Bref aperçu de la méthode

Objectifs

L'objectif est de préparer le projet : rationaliser le travail dès le début car cela se traduira finalement par un gain de temps.

De plus, fournir des esquisses (graphiques) est efficace pour présenter le projet :

- À sa hiérarchie : rationalisé, le projet n'en sera que plus convaincant.
- Aux utilisateurs : une représentation simple et compréhensible des fonctions améliorera la qualité des discussions.
- À l'équipe de développement : le rôle de chacun et les expertises attendues seront bien identifiés.

Ainsi, ce travail de préparation du projet va vous apporter :

- de l'aisance quand vous aurez à parler de votre projet ;
- de l'assurance car vous pourrez comparer différentes solutions ;
- de l'efficacité en avançant plus rapidement qu'en utilisant les méthodes classiques ;
- la possibilité de contrôler la complétude du système (n'oublier aucune fonction) ;
- la capacité à évaluer la quantité de travail de développement à réaliser ;
- un support utile pour organiser les développements ;
- une bonne maîtrise des différences entre les aspects fonctionnels et techniques du futur système.

On le voit ici, l'objectif recherché est celui de la clarification d'un problème.

Puisqu'elle permet d'étudier le système plus ou moins en profondeur (effet de zoom), la méthode laisse la liberté de le représenter de manière plus ou moins macroscopique. C'est à vous de choisir, mais n'oubliez toutefois jamais le principe des cartes routières : schématiser la réalité sans la représenter totalement ! (si une carte routière était égale à la réalité qu'elle représente, elle serait aussi grande que cette réalité...). Le but d'une telle méthode est d'obtenir un schéma compréhensible du futur système.

La phase de préparation du projet, objet de ce chapitre, vise à vous fournir le plus rapidement possible une vue schématique des fonctions et services de votre futur système.

Principes généraux

De la donnée élémentaire stockée dans une base de données à la page HTML, d'un plan de bureau d'étude à la description d'une opération de montage/démontage, des données de spécification d'emballage aux documents décrivant les conditions de transport, les exemples ne manquent pas du long et permanent processus de traitement de

l'information. Maintes fois répété, il est toujours le même : des informations brutes sont choisies pour former un ensemble informatif cohérent qui, à son tour, est transmis sous la forme de documents (des informations rendues lisibles) à des personnes (ou systèmes informatiques) capables de les interpréter et de les utiliser à bon escient.

À partir de ce constat, la méthode repose sur quatre principes élémentaires :

- Le processus d'extraction, assemblage et publication est invariant et caractérise tout système d'information.
- Au sein d'un système d'information, le composant élémentaire est un service capable de recevoir une information, la transformer et la transmettre à son tour à un autre composant élémentaire. Tout composant élémentaire est constitué d'au moins deux fonctions : l'une sert à transmettre, l'autre à effectuer un traitement.
- Information et rigidité sont deux mots antinomiques. Le maître mot est *flexibilité*.
- Pour faire face à la complexité du problème, il est impératif de pouvoir travailler sur des représentations simples : synthétiser pour mieux comprendre.

De manière plus concrète, la méthode met en évidence les fonctions du système et, *via* un mécanisme de classification et d'attribution de points, transforme une liste de fonctions en courbes et histogrammes explicites.

De par l'importance qu'elle donne aux échanges d'information, la méthode se distingue de celles fondées sur UML et l'analyse des cas d'utilisation (*use cases*) du système. Ces dernières cernent les objets à partir des différents modes d'utilisation du système étudié, chacun étant décrit à l'aide d'*un cas d'utilisation* d'UML. C'est alors en s'appuyant sur ces cas d'utilisation que le concepteur fera émerger très tôt les principes de la conception orientée objet : l'encapsulation des données et des traitements par des classes. L'accent mis dès le début sur les classes contribue alors implicitement à privilégier l'analyse des traitements plutôt que celle des données échangées. La primauté donnée au composant objet conduit ainsi à un typage des données orienté traitement qui rend plus difficile une conception orientée vers l'autonomie des données échangées, garante de plus de flexibilité et de pérennité. L'échec du déploiement à grande échelle des architectures d'objets distribuées a contribué au développement des architectures basées sur les services où la coordination entre les différentes unités de traitement, appelées « services », est réalisée à l'aide d'échanges d'information. C'est ce principe d'analyse des échanges qui sert de base à la démarche proposée dans ce chapitre.

Limites

« *L'utilité de la méthode s'arrête là où commence celle des autres* », à savoir :

- Elle n'a pas la prétention d'être une théorie générale de la modélisation des systèmes d'information. Sa seule vocation, et avantage, est d'être simple, pratique et

rapide à mettre en œuvre. Elle est limitée au travail de réalisation d'une esquisse du futur système.

- Elle ne concerne que la première des étapes permettant d'aboutir à la modélisation complète du système. D'autres suivront d'où sortiront, par exemple, des schémas XML, des modèles de traitement et de stockage. À ce titre, elle ne couvre pas les considérations relatives à la mise en œuvre physique du futur système.

Contexte idéal d'utilisation de la méthode

Cette section présente quelques définitions permettant de mieux cerner le contexte technique et conceptuel dans lequel les résultats obtenus avec la méthode sont les meilleurs.

Un *système de traitement de l'information* est un ensemble constitué d'une ou plusieurs application(s) informatique(s), capable de transformer en information des données élémentaires. Délivrer cette information à des utilisateurs est sa principale raison d'être, quelle que soit la forme physique choisie pour remettre cette information.

Le terme *information* est volontairement utilisé plutôt que *donnée*. Nous entendons par information un ensemble de données formant une *unité logique d'information*.

Une *unité logique d'information* est un ensemble de données auxquelles on a donné du sens en suivant une logique informative. L'unité logique d'information ressemble à un document, à savoir une entité ayant un sens informationnel pour l'humain qui la lit dans un contexte particulier. Cependant, cela peut également être une partie d'un document : par exemple, l'ensemble des données composant une commande, abstraction faite de celles concernant le destinataire de cette commande (et vice versa).

Les *données* peuvent être de type *unitaire*, *documentaire* ou *graphique*, c'est-à-dire qu'elles peuvent correspondre à des champs d'une base de données, des paragraphes textuels et des objets visuels. Isolée, une donnée n'a qu'un sens tandis qu'associée à d'autres, elle devient *information* et peut avoir plusieurs sens.

Présentation détaillée des quatre principes de base de la méthode

Principe n°1. Un système d'information repose toujours sur un même modèle de base.

Le premier principe de base est construit autour de l'idée que l'information est une matière vivante. À ce titre, elle a un début, une vie et une fin : elle naît, grandit et meurt.

Ce cycle de vie est simple à représenter. En termes techniques, le début d'une information se situe dans un domaine baptisé « production » tandis que sa croissance se passe dans un « vivier » que nous avons baptisé « gestion ». Quant à sa fin, nous la faisons commencer dès que l'information est diffusée ; en effet, elle échappe alors au système qui l'a vu naître et passe dans un autre monde.

C'est pourquoi la représentation la plus élémentaire de notre système d'information est constituée de trois zones, représentées à la figure 1-1.

Figure 1-1

La représentation la plus élémentaire de notre futur système d'information



C'est à partir de ce socle que nous allons pouvoir construire une première représentation de notre système, brique par brique en suivant la démarche suivante :

- identifier les fonctions du système,
- placer chaque fonction au dessus du socle qui lui convient,
- empiler les fonctions selon un plan (que nous allons vous présenter plus loin),
- construire l'ensemble en pensant que chaque fonction doit rendre un service potentiellement indépendant de tous les autres.

C'est ce dernier point qui est détaillé dans le principe n°2.

Principe n°2. Un système d'information est un ensemble de fonctions et services.

VOCABULAIRE Service

Selon les principes des architectures orientées service, un service est une unité de traitement autonome communiquant avec son environnement à l'aide de messages.

Les services définissent ainsi des frontières entre les lieux de traitement de l'information, l'intérieur des services, et les lieux d'échanges de l'information – la communication entre les services.

VOCABULAIRE Fonction

On peut définir une fonction comme étant un traitement permettant de produire un résultat spécifié à partir d'une entrée également spécifiée.

Les définitions des mots « service » et « fonction » sont très proches l'une de l'autre. À partir de la définition des services, on passe très facilement à celle de fonction. En com-

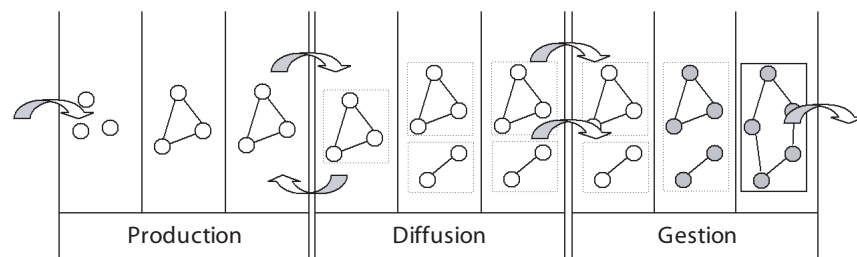
munication, la différence entre les deux est infime. En plaçant la question de la communication de l'information au cœur de la problématique de conception (architecture) des systèmes, on fait ressortir le rôle des services, à savoir les fonctions du système qui vont ingérer, transformer et traiter l'information reçue sous forme de documents XML. En ce sens, les architectures orientées services et l'échange d'information (de documents) au format XML sont deux composantes non seulement complémentaires mais consubstantielles des systèmes d'information objets de cet ouvrage.

Vous allez voir que, grâce à ce principe, on peut obtenir une bonne représentation des échanges d'information au sein du futur système. La mise en valeur des communications entre les services appartenant aux différentes zones du socle illustré à la figure 1-1 – production, gestion, diffusion – met en évidence les étapes les plus significatives des différentes phases de traitement et d'enrichissement de l'information. Nous verrons ultérieurement qu'il existe une relation de cause à effet directe entre la qualité avec laquelle ces communications inter-zones sont identifiées et la flexibilité finale du système. Cela implique une relation de cause à effet directe entre la qualité de cette conception et le coût de développement et maintenance du système final.

Dans la figure 1-2, le système est représenté comme s'il s'agissait d'une usine de traitement de l'information : à gauche y entrent des granules d'information, des données, et à droite sort un produit raffiné qui est bien souvent un document.

Au dessus de la brique « production » du socle vont se retrouver toutes les fonctions qui servent à créer ou récupérer les granules d'information initiaux.

Figure 1-2
Fonctions et services
composent un système
d'information.



Au dessus de la brique « gestion » seront représentées toutes les fonctions d'assemblage, de gestion de la configuration et des processus de fabrication d'une information finie.

Au dessus de la brique « diffusion » seront empilées toutes les fonctions permettant de diffuser l'information. Dans la figure 1-2, nous essayons de montrer la logique avec laquelle nous allons construire une représentation efficace de notre futur système, mêlant petit à petit les notions de fonctions et de services. De granules blancs et isolés les uns des autres à l'entrée du système, on montre comment ils sont assemblés, placés par rapport à un contexte extérieur dans la brique « gestion » (contexte

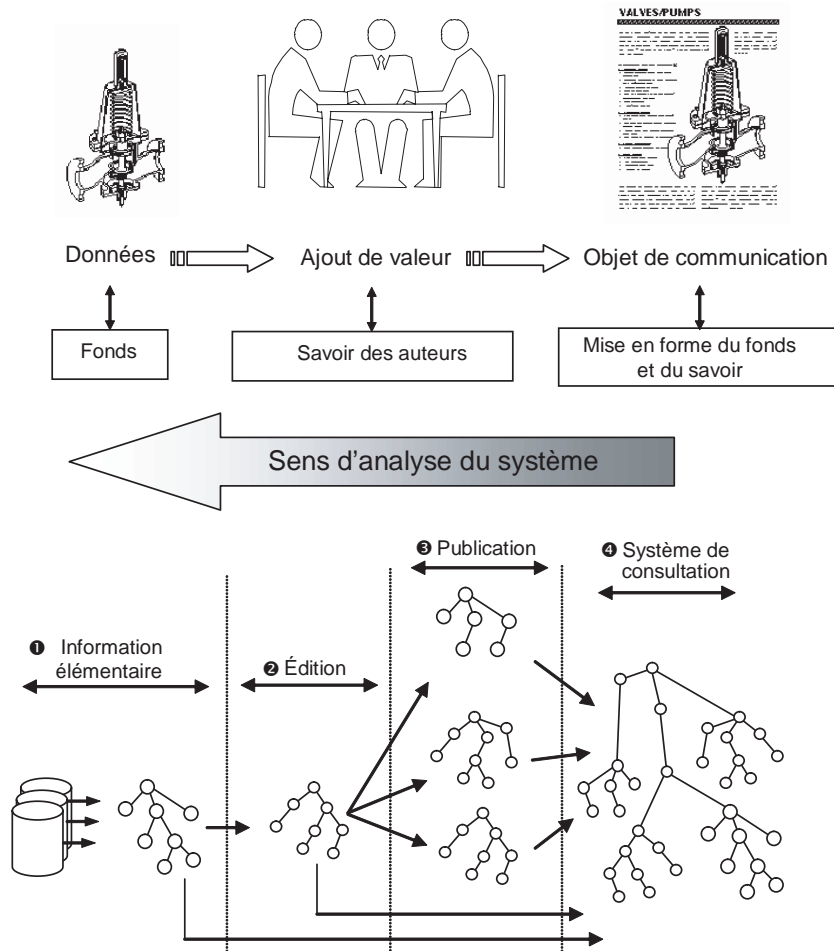
symbolisé par un rectangle en pointillé), puis comment ils sont adaptés à un contexte de diffusion (les granules blancs deviennent gris).

Le tout respecte le troisième principe, celui de la flexibilité, décrit dans la prochaine section.

Principe n°3. Un système d'information est obligatoirement flexible.

L'information, presque par définition, est en évolution permanente. Au cours du temps cependant, les traitements changent à un rythme encore plus rapide que les données. Ces dernières doivent donc être organisées de manière à assurer au système une grande flexibilité dans le temps.

Figure 1-3
L'information DOIT être articulée et transportable.



Si vous respectez le principe n°2, vous aurez à l'esprit que votre système manipulera des granules éminemment transportables alors que dans les méthodes de conception classiques, on a plutôt tendance à figer l'information dans des cases...

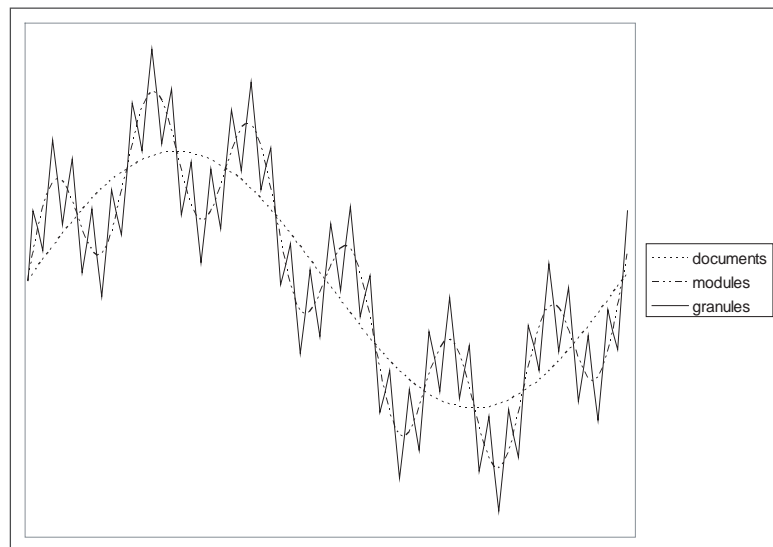
La flexibilité d'un système d'information équivaut aux articulations du squelette humain : elle confère au système, non des possibilités de mouvement, mais d'adaptation de l'information. Elle permet d'utiliser l'information à plusieurs fins, sur plusieurs médias, avec différents traitements de transformation. La figure 1-3 schématise plusieurs idées relatives à la flexibilité :

- Le système doit s'adapter (en permanence) à l'évolution des organisations humaines.
- Le système doit s'adapter aux besoins des consommateurs de l'information et de l'évolution du marché qu'ils représentent.
- Le système doit permettre de réutiliser une même information, ou granule d'information, ou donnée élémentaire.

Enfin, la figure 1-4 schématise un autre type de flexibilité : celle du rythme des modifications des informations gérées par le système. Certaines informations ou données seront modifiées très régulièrement (des cours de bourse par exemple), alors que les documents qui contiendront ces données seront potentiellement mis à jour selon un rythme différent. Toujours en prenant l'exemple traditionnel des données boursières, pensons aux documents papier contenant des analyses financières ou les pages Web qui affichent les cours : on accepte la mise à jour régulière de la page, mais on n'accepte pas que sa forme change continuellement.

Figure 1-4

Le système doit permettre de prendre en compte les fréquences de mise à jour propres à chaque type d'information.

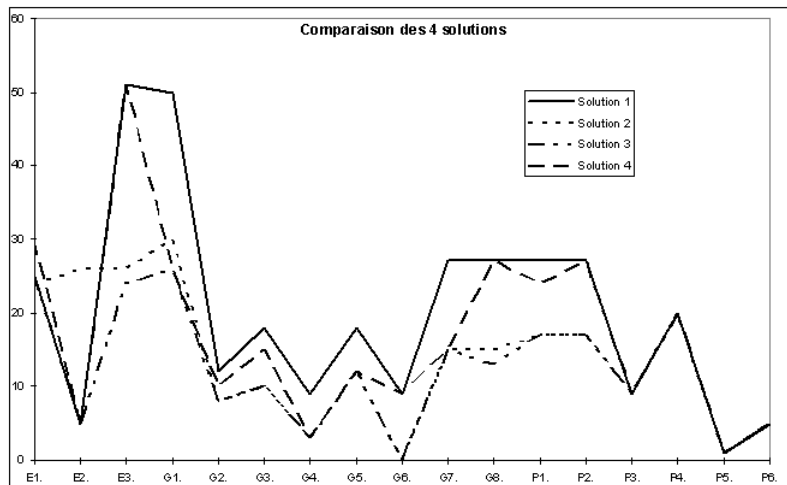


Principe n°4. La représentation du système d'information doit être synthétique et simple.

La représentation graphique du système repose sur des règles simples. Cela facilite la comparaison de différentes solutions. Cette simplicité est donc un but recherché. Avec une représentation trop complète ou trop sophistiquée, cette comparaison serait impossible : le seul temps passé à établir de telles représentation ferait perdre l'intérêt même de la comparaison. Un exemple typique est celui des tarifs des opérateurs de téléphonie : à vouloir être trop exhaustifs dans leurs tarifs, toute comparaison globale devient impossible ! C'est ce que nous voulons éviter avec notre méthode.

D'après la méthode, le système va être progressivement découpé en services (ensembles de fonctions de traitement) et échanges de données entre services. Un système simple d'attribution de points permet de noter chaque fonction et chaque service. La grille d'attribution des points ainsi obtenue permet d'associer un coût théorique à chaque solution étudiée : plus le nombre de points est élevé et plus la solution sera coûteuse à réaliser. La valeur des points aide ainsi à se faire une idée du coût de réalisation du projet.

Figure 1-5
Courbes représentant
le système au terme de la
phase d'ébauche



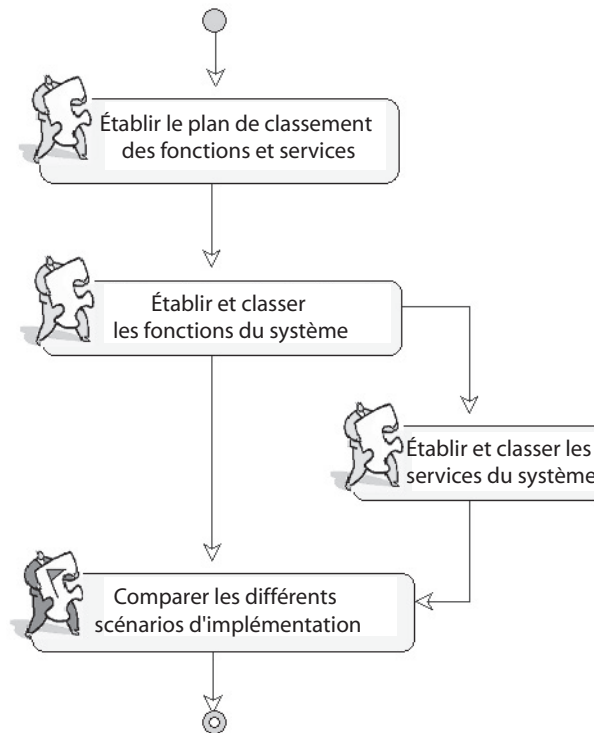
Au terme de sa mise en œuvre, la méthode produit des courbes parlantes, représentant le système ébauché. Par exemple, la figure 1-5 fournit le graphique obtenu au terme de la pré-étude d'un système qui devait assurer la gestion de configuration entre des systèmes mécaniques ayant de multiples options et la documentation décrivant justement lesdites options. Pour cette étude, quatre hypothèses technologiques différentes avaient été retenues. Les courbes obtenues avec la méthode proposée dans ce chapitre ont permis de comparer en un clin d'œil le coût respectif de chacune de

ces hypothèses. Plus précisément, la représentation a permis de comprendre et discuter chaque poste de coût, fonction par fonction, service par service.

Mise en œuvre de la méthode

La mise en œuvre de notre méthode se déroule en trois ou quatre phases selon la profondeur d'analyse recherchée (figure 1-6). Dans le cas de systèmes simples, l'étape d'analyse des services peut être omise. Dans les systèmes plus complexes, l'identification des services, c'est-à-dire des canaux de communication, donne le moyen de maîtriser les flux d'information. Cette étape devient carrément indispensable dans le cas des systèmes complexes ayant de nombreuses interactions avec d'autres systèmes.

Figure 1-6
Représentation des différentes étapes de la mise en œuvre de la méthode



Au stade ultime de la méthode, on peut établir des tableaux comparatifs de différentes solutions techniques. En fin de chapitre, la présentation d'un cas réel en donnera un exemple.

Étape n°1. Établir un plan de classification des fonctions et services du système

Comme nous l'avons abordé dans les premières sections de ce chapitre, la définition la plus élémentaire d'un système d'information est constituée des trois zones production, gestion et diffusion. Invariablement, le cycle de vie de toute donnée, fichier ou document s'inscrit dans les fonctions et services qui s'y trouvent définis.

La zone de production regroupe les fonctions associées à la naissance de l'information. Cela peut être le résultat d'une importation depuis un autre système, d'une conversion, d'une génération spontanée consécutive à un calcul, ou encore d'une action manuelle humaine comme la saisie d'une donnée ou d'un document.

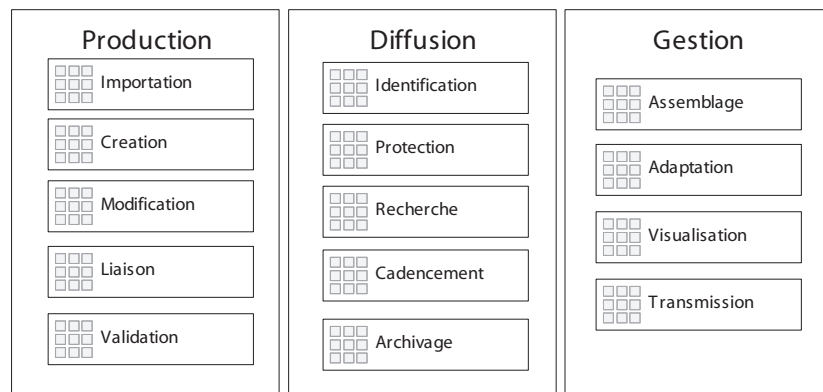
La zone de gestion regroupe les fonctions associées au stockage de l'information dans le système où elle sera contrôlée, protégée, mise dans un processus et archivée. Cette zone sert à garantir la qualité et la pertinence de l'information. Pour cela, une fonction importante de la zone est la gestion de configuration : le système doit permettre de garantir la cohérence des informations les unes par rapport aux autres.

La zone de diffusion regroupe les fonctions qui servent à préparer l'information aux médias de diffusion, en faire des pages HTML ou papier ou transmettre l'information à un autre système.

Chaque zone est composée d'un certain nombre de fonctions, regroupées en blocs. La méthode propose une série de blocs standards, ce qui ne vous empêche pas ultérieurement d'ajouter les vôtres. Ceux que nous proposons sont très utiles lors du démarrage de l'étude de conception d'un système.

La figure 1-7 présente les blocs standards de la méthode, ceux que l'on retrouve inmanquablement dans tout système d'information

Figure 1-7
Les blocs fonctionnels standards de la méthode



Pour la zone *production*, les blocs standards sont les suivants :

- *Importation* : fonctions de récupération des données d'un autre système, logiciel ou base de données.
- *Création* : fonctions de création des objets de type donnée, texte, image, illustration etc.
- *Modification* : fonctions d'édition des objets créés dans les blocs *création* et/ou *importation*.
- *Liaison* : fonctions de pose de liens entre les objets.
- *Validation* : fonctions de contrôle de conformité des objets déposés dans le système d'information.

Pour la zone *gestion*, les blocs standards sont les suivants :

- *Identification* : fonctions d'étiquetage des objets stockés dans le système. Ces données d'identification servent la plupart du temps à retrouver, classer et gérer les objets contenus dans le système. Dans le monde des documents, les données d'identification sont souvent appelées *méta-données*.
- *Protection* : fonctions de contrôle d'accès aux objets par la définition de droits. Des programmes, autant que des humains, sont susceptibles d'exploiter les objets se trouvant dans le système. Les droits d'accès peuvent changer en fonction d'événements divers : une même personne, ou un programme, peut voir ses droits évoluer en fonction de la propre évolution de l'objet auquel accéder.
- *Cadencement* : fonctions de définition et d'activation des processus composant le cycle de vie des objets, tous susceptibles d'être soumis à des circuits de validation particuliers avant d'être publiés.
- *Recherche* : fonctions de recherche et d'analyse des objets de la base pour en garantir l'intégrité ou tout simplement les retrouver.
- *Archivage* : fonction de recopie (voire destruction) des objets du système pour en faire des copies de sécurité ou les rendre hors d'usage tout en conservant la trace.

Pour la zone *diffusion*, les blocs standards sont les suivants :

- *Assemblage* : fonctions d'agrégation et collecte des objets afin d'en faire un tout cohérent pour un média de restitution et un objectif d'information.
- *Adaptation* : fonction de transformation des objets pour les adapter non seulement aux médias de diffusion mais encore aux lecteurs ou systèmes informatiques destinataires de l'information.
- *Visualisation* : fonctions d'affichage des documents obtenus par les fonctions d'assemblage et adaptation.
- *Transmission* : fonction de transport des documents obtenus jusqu'à leurs destinations finales.

C'est en adaptant ce plan générique « d'urbanisation » au cas de votre propre système d'information que vous allez peu à peu le structurer et trouver ses spécificités.

Étape n°2. Identifier et classer les fonctions du système

Dans les projets informatiques, il est classique d'analyser les fonctions et sous-fonctions d'un système. Cette étape repose tant sur les demandes des utilisateurs que sur le propre savoir du concepteur du système. En effet, si les utilisateurs connaissent leurs besoins, ils ne sont que rarement capables de les mettre en regard d'une logique informatique. Il revient au concepteur de faire ce travail d'appairage.

Pour conduire cette étape, il faut partir des demandes fonctionnelles des utilisateurs. On établit alors une liste de fonctions et sous-fonctions, si possible en limitant à 1 le nombre de sous-niveaux de fonctions (interdisez-vous d'avoir des sous-sous-fonctions).

Une fois cette liste établie (le tableau 1-1 en donne un exemple), chaque fonction ou sous fonction est associée à l'une (et une seule) des trois zones élémentaires du système. Dans la colonne *zone*, on met un P quand la fonction correspond à la zone *production*, un G quand il s'agit de la zone *gestion* et un D pour la zone *diffusion*.

Dans le tableau, vous observerez également que chaque fonction ou sous-fonction se voit attribuer un code d'identification (FP13 par exemple pour la fonction « créer une structure de document en XML »).

Le tableau 1-1 fournit un exemple du résultat à obtenir pour trois des fonctions principales d'un système. Il ne s'agit que d'un extrait, un système réel étant composé de plusieurs dizaines de fonctions principales.

À la manière de XML Schema, qui ne fournit pas un schéma universel mais est un outil pour créer tout type de modèle, notre méthode ne fournit pas une décomposition exhaustive d'un système. C'est en ce sens qu'elle n'a pas la prétention d'être une théorie générale de l'architecture d'un système d'information. Notre méthode est un outil pour construire votre propre représentation de votre système : à vous d'identifier les fonctions principales et secondaires qui caractériseront votre système selon le plan générique que nous fournissons et qui, quant à lui, est invariant.

Tableau 1-1 Exemple de fonctions identifiées lors d'une étude préalable de besoin

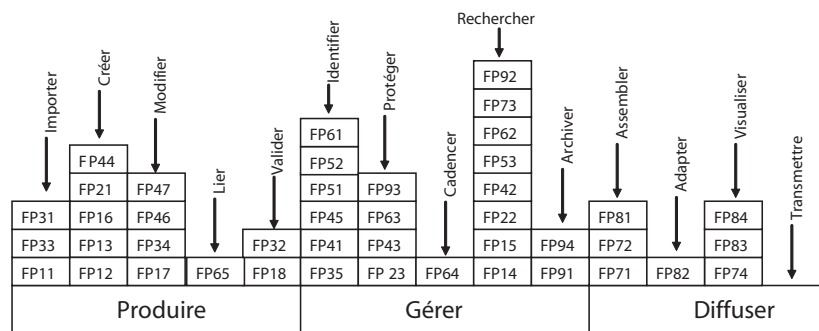
Réf. interne	Identification des fonctions	Zone
FP1	Manipuler des structures de données et produire les fichiers XML correspondants.	
FP11	Importer dans la base une structure de données définie extérieurement.	P
FP12	Créer des squelettes de publication.	P
FP13	Créer une structure de document en XML.	P

Tableau 1-1 Exemple de fonctions identifiées lors d'une étude préalable de besoin (suite)

Réf. interne	Identification des fonctions	Zone
FP14	Comparer (rechercher et analyser) deux structures de données.	G
FP15	Connaître les impacts de l'évolution d'une structure (rechercher et analyser).	G
FP16	Créer des structures de données de test.	P
FP17	Modifier des squelettes de publication.	P
FP18	Valider les unités d'information.	P
FP2	Gérer des structures de publication.	
FP21	Créer des modèles réutilisables de structures de publication.	P
FP22	Rechercher des structures de publication.	G
FP23	Contrôler les accès aux différentes fonctions relatives à la gestion des structures de publication.	G
FP3	Manipuler des unités d'information et des graphiques.	
FP31	Disposer d'un programme permettant d'introduire des illustrations dans la base.	P
FP32	Valider les unités d'information entrant dans la base. Pour les unités d'information XML, il s'agit d'utiliser un parseur. En cas d'erreur, un opérateur pourra intervenir.	P
FP33	Convertir en XML les unités d'information non-XML. Convertir les illustrations aux standards retenus.	P
FP34	Éditer des unités d'information.	P
FP35	Disposer d'un programme permettant d'introduire dans la base des unités d'information.	G
...

Ce travail étant fait, l'étape suivante consiste à reporter chaque fonction sur un histogramme (figure 1-8) qui va fournir une première représentation visuelle du système. Ce placement est fait selon une logique que nous expliquons dans la suite de cette section.

Figure 1-8
Représentation sous forme d'histogramme des fonctions du système



Les fonctions sont placées en respectant la logique suivante :

- Le socle de base de l'histogramme représente les zones.
- Au dessus de chaque zone sont posées les fonctions identifiées au tableau 1-1 : une brique par sous-fonction. Chaque colonne correspond à l'un des blocs définis pour chaque zone (*importer, créer, modifier etc.*).
- S'il se trouve des fonctions d'interfaçage entre les zones, telles que *production et gestion*, elles seront placées de part et d'autre de la ligne verticale qui sépare les zones concernées. Lorsque cela est possible, les fonctions qui se répondent seront placées en regard les unes des autres. Par exemple, les fonctions FP18 (*validation des unités d'information*) et FP35 (*programme d'importation des unités d'information validées*) se correspondent dans notre exemple (en général, c'est le concepteur ayant établi la liste des fonctions qui peut le dire) : elles sont donc placées en regard l'une de l'autre.
- L'ordre vertical n'est pas imposé par la méthode : choisissez votre propre règle si cela permet à votre modèle d'être encore plus explicite. Vous pouvez par exemple les positionner en fonction de leur ordre de dépendance les unes avec les autres, ou encore en mettant celles qui vous semblent les plus importantes en bas et en terminant en haut par d'éventuelles options du système.
- À l'extrême gauche de l'histogramme se trouvent les fonctions d'importation dans le système tandis qu'à l'extrême droite sont placées les fonctions de diffusion vers l'extérieur. Cela permet d'avoir une compréhension rapide du graphe : à gauche sont représentées toutes les tâches nécessaires pour alimenter le système et à droite toutes celles pour émettre l'information. Au centre se trouvent les fonctions de gestion.

Étape n°3. Identifier et classer les services du système

Cette étape concerne les systèmes de bonne taille. Quand le système est simple, l'analyse fonctionnelle peut suffire et cette étape est alors inutile.

L'analyse fonctionnelle traditionnelle ne met que rarement en évidence les interactions ayant lieu tant à l'intérieur du système qu'entre le système et son environnement. En effet, cette question est essentiellement technique et non fonctionnelle. Le but de l'étape 3 est justement d'identifier ces interactions car elles conditionneront l'organisation des composants logiciels, ou services dans ce cas, du système. Les équipes techniques pourront alors développer une architecture *orientée service* du système sans perdre de vue son objectif fonctionnel.

Les services sont des unités autonomes de traitement, coordonnées par des messages. Ils organisent les fonctions découvertes lors de l'étape précédente pour faire apparaître les canaux de communication du système. L'organisation des services, c'est-à-

dire la manière dont ils vont intervenir dans la vie du système, représentera la manière dont les fonctions interagissent entre elles, en particulier en ce qui concerne les données ou objets échangés. Les services vont ainsi dévoiler un aspect très important de la conception : le format, ou modèle, des données échangées entre les fonctions du système. Il y a ainsi une étroite relation entre les services et la conception des modèles de données.

En complément de l'analyse purement fonctionnelle, l'identification des services revient à bien isoler tous les points de transformation ou d'échange d'objets (principalement des données ou des informations). Elle complète et organise utilement la liste des fonctions identifiées lors de la phase précédente.

Une attention particulière doit être portée aux services situés à la frontière entre les zones de production, de gestion et de diffusion. Les échanges entre ces services représentent les points d'articulation les plus sensibles du système. Leur bon recensement est une condition requise pour l'élaboration de scénario d'implémentation de qualité.

Étape n°4. Comparer les scénarios d'implémentation

La dernière partie de la méthode va aboutir au chiffrage des différentes solutions d'implémentation. Pour cela, nous vous proposons une démarche qui vous permettra d'attribuer des points aux fonctions et services identifiés lors des étapes précédentes.

Pour que l'analyse du système soit objective, les points seront choisis en fonction de critères objectifs !

L'analyse va consister à donner à chaque fonction et service du système étudié une série de cinq notes qui représentent :

- 1 le poids fonctionnel (l'acronyme utilisé, FW, est celui de *Function Weight*) ;
- 2 le poids logiciel (ou SW comme *Software Weight*) ;
- 3 le poids du développement (ou DW comme *Development Weight*) ;
- 4 le coût du développement (ou DC comme *Development Cost*) ;
- 5 le coût du développement avec prise en compte de la qualité (ou DCQ comme *Development Cost with Quality*).

Nous allons voir maintenant comment appliquer les notes.

Le *poids fonctionnel* est une note qui vaut 0 quand la fonction est optionnelle ou inutile, et 1 quand elle est impérative. Il ne s'agit pas uniquement de mesurer par là l'intérêt d'une fonction par rapport au système cible, mais de faire apparaître des différences entre plusieurs solutions techniques. Par exemple, il se peut qu'une fonction de transformation de données soit nécessaire dans le cas d'une solution technique et inutile dans l'autre.

Le *poids logiciel* est une note qui peut prendre les valeurs 0, 1 ou 2. La note 0 est attribuée quand la fonction est optionnelle ou inutile et que son poids fonctionnel est déjà 0. La note 1 est attribuée aux fonctions mises en œuvre, de base, dans la solution technique concernée. La note 2 signifie que la fonction n'est pas de base dans la solution technique envisagée et doit faire l'objet d'un paramétrage ou d'un développement spécifique.

Le *poids du développement* est une note comprise entre 0 et 3 : la note 0 est conservée pour les fonctions optionnelles ou inutiles, la note 1 pour les fonctions de la solution étudiée, la note 2 correspond aux fonctions qui peuvent être mises en œuvre par un simple paramétrage tandis que la note 3 correspond aux fonctions qui nécessitent un développement spécifique.

Le *coût du développement* est l'évaluation du temps nécessaire pour développer les fonctions dont le poids du développement est 3. Pour les autres fonctions, la note correspondant au poids du développement est maintenue.

Enfin, le poids de la qualité est le poids du développement augmenté d'un coefficient qualité multiplicateur à choisir en fonction du niveau de qualité exigé pour l'application et des habitudes de votre société sur ce point.

Les tableaux 1-2 et 1-3, ainsi que la figure 1-9, illustrent les résultats obtenus. Le tableau 1-2 présente la liste comparative des fonctions identifiées pour les besoins d'un système de traitement de l'information pour lequel quatre solutions techniques – baptisées sol1, sol2, sol3 et sol4 – ont été étudiées. Ce tableau est un extrait d'une étude réelle complète portant sur une centaine de fonctions.

Les colonnes de droite du tableau contiennent les 0 et 1 du poids de chaque fonction dans chacune des solutions envisagées.

Tableau 1-2 Liste comparative des fonctions de quatre solutions

Liste des fonctions de chaque solution							
Zone	MF	SF	Description	Sol1	Sol2	Sol3	Sol4
PRODUCTION							
	E1		Importer				
		FP11	Importer dans la base une structure de données définie extérieurement.	1	1	1	1
		FP33	Convertir en XML les unités d'information non-XML. Convertir les illustrations aux standards retenus.	1	1	1	1
		FP31	Disposer d'un programme permettant d'introduire dans la base des unités d'information.	1	1	1	1
	E2		Créer				
		FP12	Créer des squelettes de publication.	1	0	1	1
		FP13	Créer une structure de données en XML.	1	0	1	1

Tableau 1-2 Liste comparative des fonctions de quatre solutions (suite)

Liste des fonctions de chaque solution								
Zone	MF	SF	Description	Sol1	Sol2	Sol3	Sol4	
GESTION								
	G1		Identifier					
		FP35	Disposer d'un programme permettant d'introduire des illustrations dans la base.	1	1	1	1	
		FP41	Gérer les unités d'information et les graphiques associés.	1	1	1	1	
		FP45	Gérer les chemins d'accès aux unités d'information qui seraient stockées à l'extérieur de la base.	1	1	1	1	
	G2		Protéger					
		FP23	Contrôler les accès aux différentes fonctions relatives à la gestion des structures de publication.	1	1	1	1	
		FP43	Contrôler les accès aux unités d'information (notion de workflow).	1	1	1	1	
		FP63	Contrôler les autorisations d'accès aux fiches d'identité.	0	1	1	1	
		FP93	Pouvoir mettre les publications dans une zone protégée du système.	0	0	0	1	
	G3		Cadencer					
		FP64	Gérer les processus.	1	1	1	1	
	DIFFUSION							
		P1		Assembler				
		FP71	Produire une structure de données en une instance XML contenant l'ensemble des textes.	1	1	1	1	
		FP72	Pouvoir effectuer cet export selon différents critères de sélection.	0	1	1	1	
		FP81	Pouvoir choisir parmi plusieurs formats de présentation.	0	0	1	1	
P2		Adapter						
		FP82	Disposer des convertisseurs XML->papier.	1	1	1	1	
P3		Visualiser						
		FP74	Visualiser des unités d'information ou des graphiques depuis l'interface de gestion de la base.	1	1	1	1	
		FP83	Disposer d'un navigateur.	0	1	1	1	
		FP84	Prévoir un poste de contrôle des versions papier ou électroniques produites.	1	1	1	1	
P4.		Transmettre						
...		
...		
TOTAUX				28	31	36	41	

Le tableau 1-3 représente toutes les notes obtenues par l'une des solutions étudiées (à savoir la solution 3). L'on y retrouve, pour chaque fonction : son poids fonctionnel (colonne FW), son poids logiciel (colonne SW), le poids du développement (colonne DW), le coût du développement (colonne DC), et enfin le poids de la qualité (colonne DCQ).

Tableau 1-3 Tableau d'évaluation de la solution n°3

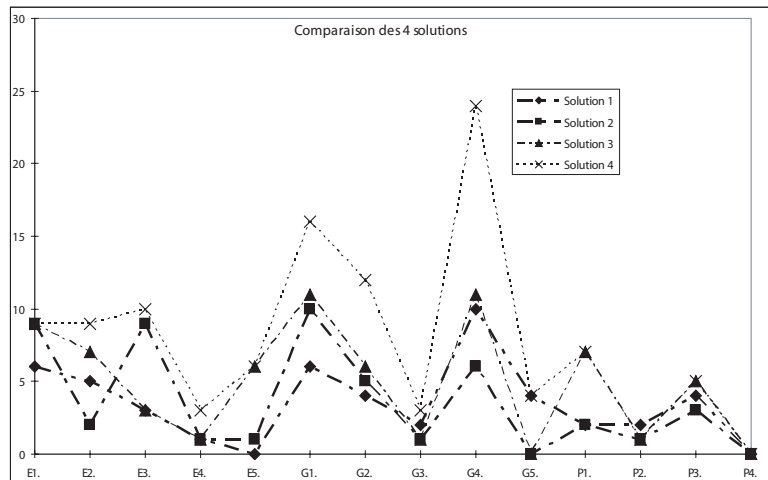
SOLUTION 3									
Zone	MF	SF	Description	FW	SW	SDW	DC	DCQ	
PRODUCTION									
	E1	Importer							
		FP11	Importer dans la base une structure de données définie extérieurement.	1	2	3	7	8,89	
		FP33	Convertir en XML les unités d'information non-XML. Convertir les illustrations aux standards retenus.	1	2	3	7	8,89	
		FP31	Disposer d'un programme permettant d'introduire dans la base des unités d'information.	1	2	3	5	6,35	
		E2	Créer						
			FP12	Créer des squelettes de publication.	1	1	1	1	1,27
		FP13	Créer une structure de données en XML.	1	2	3	3	3,81	
GESTION									
	G1	Identifier							
		FP35	Disposer d'un programme permettant d'introduire des illustrations dans la base.	1	2	2	2	2,54	
		FP41	Gérer les unités d'information et les graphiques associés.	1	1	1	1	1,27	
		FP45	Gérer les chemins d'accès aux unités d'information qui seraient stockées à l'extérieur de la base.	1	1	1	1	1,27	
		G2	Protéger						
			FP23	Contrôler les accès aux différentes fonctions relatives à la gestion des structures de publication.	1	2	2	0,5	0,635
			FP43	Contrôler les accès aux unités d'information (notion de workflow).	1	1	1	0,5	0,635
			FP63	Contrôler les autorisations d'accès aux fiches d'identité.	1	2	3	3	3,81
			FP93	Pouvoir mettre les publications dans une zone protégée du système.	0	0	0	0	0
		G3	Cadencer						
			FP64	Gérer les processus.	1	1	1	0,5	0,635
	DIFFUSION								
	P1	Assembler							
		FP71	Produire une structure de données en une instance XML contenant l'ensemble des textes.	1	1	1	1	1,27	

Tableau 1-3 Tableau d'évaluation de la solution n°3 (suite)

SOLUTION 3						FW	SW	SDW	DC	DCQ
Zone	MF	SF	Description							
		FP72	Pouvoir effectuer cet export selon différents critères de sélection.			1	2	3	3	3,81
		FP81	Pouvoir choisir parmi plusieurs formats de présentation.			1	2	3	3	3,81
	P2	Adapter								
		FP82	Disposer des convertisseurs XML->papier.			1	1	1	0	0
	P3	Visualiser								
		FP74	Visualiser des unités d'information ou des graphiques depuis l'interface de gestion de la base.			1	1	1	0	0
		FP83	Disposer d'un navigateur.			1	2	3	3	3,81
		FP84	Prévoir un poste de contrôle des versions papier ou électroniques produites.			1	1	1	1	1,27
	P4.	Transmettre								
		
		
		
		
			TOTAUX			36	53	68	85	108

En reportant ces chiffres sur une courbe, on obtient finalement une représentation dont un exemple est fourni par la figure 1-9. Les différences entre solutions apparaissent alors de manière visible et synthétique. Comparer les solutions en étudiant les raisons des différences de coût point par point devient, dès lors, d'une grande simplicité.

Figure 1-9
Courbes comparatives
des quatre solutions
de notre exemple



On retiendra pour conclure qu'une telle représentation parle tant aux techniciens qu'aux futurs utilisateurs ou décideurs. Elle est un outil de dialogue particulièrement adapté.

En résumé...

La méthode que nous avons présentée permet tout à la fois de structurer la démarche projet et de représenter de manière parlante les fonctions du futur système, tout en fournissant des indices utiles quant aux coûts respectifs des développements, au nombre de fonctions « natives », et aux expertises nécessaires au développement.

Le découpage fonctionnel donne une première idée de l'architecture des composants logiciels à mettre en place tandis que la mise en évidence des services renseigne dès le départ sur les flux de données internes et externes au système.

Cette méthode permet donc de placer un certain nombre de bases qui sont autant de fondations du futur système.

Appliquée préalablement à un choix définitif de solution, elle a l'avantage de répertorier très précisément les réelles fonctions du futur système, sans se limiter aux seules fonctions vues des utilisateurs. Les fonctions coûteuses sont rapidement identifiées. Les courbes permettent donc aux décideurs de faire des choix de mise en œuvre. Les chefs de projet ont un outil qui leur facilite les explications relatives à tel ou tel coût de développement.

Au terme de l'application de cette méthode, vous apercevez les premières relations liant les données aux fonctions du système, c'est-à-dire les grandes masses pour lesquelles des modèles de données devront être développés, cela avant même de vous être engagé dans une quelconque étude détaillée.