

Chapitre I

Problématique de la conception

1. Cas de la réalisation d'un ouvrage sur mesure

La réalisation d'un ouvrage sur mesure met en évidence la problématique de la communication entre un maître d'ouvrage qui sait ce qu'il veut et un maître d'œuvre chargé de le réaliser. L'un exprime ses idées dans un langage courant donc souvent ambiguë, imprécis et sujet à de nombreuses interprétations mais significatif pour lui. L'autre maîtrise les techniques pour la réalisation mais doit interpréter le besoin, pour réaliser la fabrication du produit désiré. Cette différence est évidemment une source importante d'erreurs, et naît de la difficulté de faire communiquer les deux mondes. Pour pallier cette difficulté, une solution consiste à décomposer tout processus de fabrication en 2 étapes :

1. **La modélisation** : elle consiste à traduire le besoin en une image fidèle du réel : *le modèle*.
2. **La réalisation** : elle consiste à fabriquer le produit, à partir du modèle.

Ce choix méthodologique fait jouer au modèle le rôle d'une expression intermédiaire du besoin entre le maître d'ouvrage et le maître d'œuvre.

Initialement, le besoin est exprimé dans le langage métier du maître d'ouvrage, voire en langage vulgaire. Les caractéristiques principales de ce langage sont l'abondance de son vocabulaire et de sa syntaxe, l'existence de synonymes, d'homonymes, qui permettent de nombreuses nuances. Cette caractéristique est une source de richesse sémantique, mais aussi d'imprécisions, de confusions, incompatibles avec la fabrication d'un produit technique. Le besoin, exprimé ainsi, ne peut pas être exploité par le maître d'œuvre. Ce dernier utilise au contraire un langage professionnel, fini et contraint, dont le rôle principal est de limiter au maximum les imprécisions et les aléas.

L'étape de modélisation réalise une reformulation du besoin en utilisant un langage intermédiaire compatible en amont avec le langage métier et en aval avec le langage du technicien. Sa caractéristique principale est l'aspect fini de son vocabulaire et de sa syntaxe. A ce titre le langage est adapté au langage du technicien. Il possède aussi la capacité démontrée, de traduire la réalité du besoin sans distorsion, sans erreur ni ambiguïté.

Il est l'interface entre deux mondes : celui du réel et celui du métier du maître d'œuvre.

Comment est-ce possible ?

L'aspect fini du langage de modélisation (vocabulaire textuel et graphique, syntaxe etc.) permet de réduire l'expression du besoin aux seuls éléments nécessaires à la fabrication du produit. Il y aura ainsi des langages de modélisation adaptés à chaque technique utilisée dans le processus de fabrication.

Pourquoi ne pas exprimer dès le départ le besoin dans ce langage ?

Ceci supposerait de la part du maître d'ouvrage une double compétence difficile à posséder en général.

Le schéma 1 montre le rôle de la modélisation dans le processus de la fabrication d'un produit.

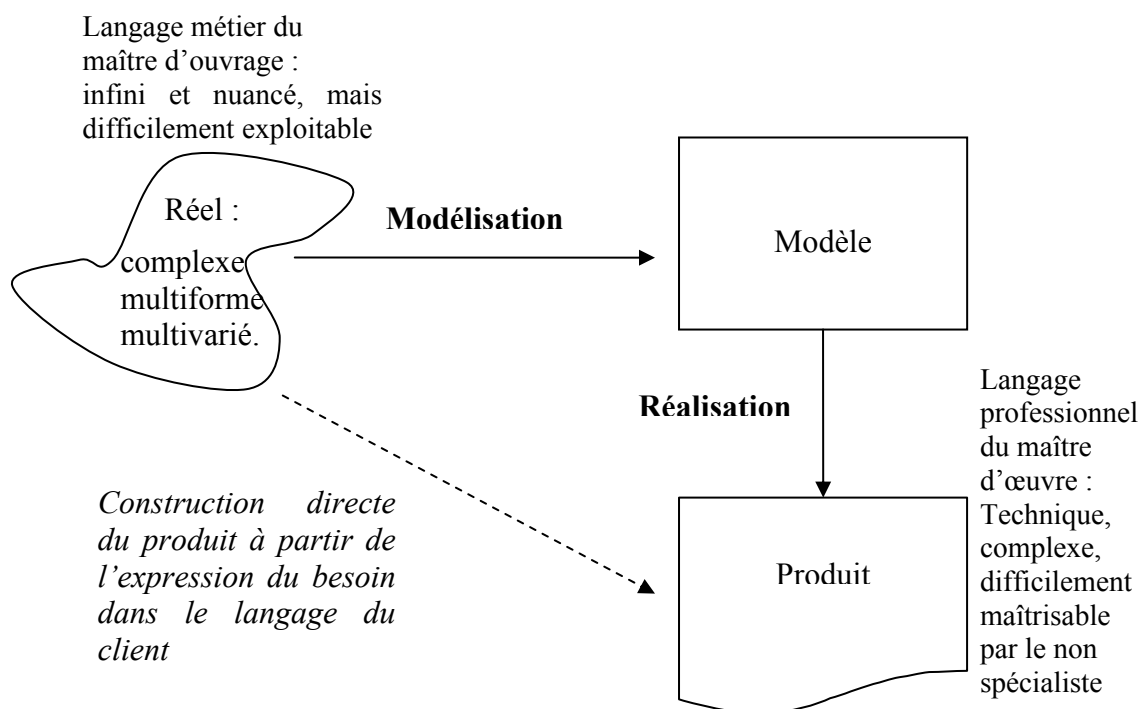


Schéma 1

Le processus de modélisation dans la fabrication d'un produit sur mesure

Pour illustrer cette démarche, analysons les flux de travaux nécessaires à la construction d'une maison. Le client exprime son besoin, dans son langage propre avec des termes pas forcément techniques, mais qui traduisent de façon précise son désir. Ceci correspond dans le schéma 1 au *réel*. A partir de ce réel exprimé, l'architecte chargé de la conception va réaliser une étude qui conduira à la fourniture d'un dossier contenant un

descriptif détaillé associé à une liasse de plans. Ce descriptif, outre son aspect contractuel, sert de base à la consultation des entreprises pour établir leurs devis.

Dans notre schéma, le descriptif détaillé et la liasse de plans correspondent au *Modèle*. Contrairement au réel exprimé dans un langage commun, et compte tenu de sa destination, le modèle est exprimé dans le langage fini des professionnels chargés d'intervenir dans la construction. Chaque corps de métier possède, dans le descriptif détaillé et dans les plans, toutes les informations nécessaires à la spécification de son intervention : une vue du modèle propre à son métier. Il y aura un chapitre pour le maçon, un pour l'électricien ... Chacun de ces chapitres constitue pour le corps d'état correspondant, le modèle nécessaire à la fabrication de la maison. Pour chacun, le vocabulaire utilisé est fini, précis, technique et permet de lever toutes les ambiguïtés de l'expression initiale du désir du client.

Par exemple, si le client désire des murs de couleur bleue, le dossier définira la couleur par rapport à une référence choisie dans un nuancier spécifique d'une marque et d'une technique utilisée par le peintre.

L'étape de modélisation permet l'appropriation du réel par le maître d'œuvre.

Il faut insister sur le rôle fondamental du langage dans cette étape. En effet, outre le travail d'innovation pour résoudre certains problèmes techniques posés par le projet, le rôle de l'architecte est celui d'un traducteur : à partir de l'expression des idées en langage vulgaire, il exprime selon le langage professionnel des gens de chaque corps d'état les désirs du client. Une fois le modèle établi, celui-ci sert de référence pour la construction de la maison.

2. Application au cas des bases de données relationnelles

La réalisation d'une base de données relève d'un processus similaire à celui qui vient d'être décrit.

Une base de données constitue la mémoire permanente de toute application informatique, et plus généralement du système d'information. Plusieurs techniques de mémorisations ont été mises en œuvre depuis que ce concept a vu le jour au milieu des années 70. Parmi elles, les bases de données relationnelles constituent la solution élégante et souple qui nous intéresse ici.

Lors de la réalisation d'une base de données relationnelle, le technicien est confronté, à un problème de langage tel qu'il a été introduit au paragraphe précédent.

Coté utilisateurs, on exprime un besoin global dans lequel les nécessités en matière de mémorisation d'informations sont *implicitement* contenues dans l'expression des besoins fonctionnels de l'application et du système d'information à mettre en place.

Coté réalisateur, l'approche relationnelle exige une séparation nette entre ce qui relève de la mémorisation des informations et ce qui est du ressort des besoins fonctionnels.

Par ailleurs, il existe des spécificités propres aux techniques de mémorisation dans une approche relationnelle que l'utilisateur final ne soupçonne pas quand il exprime son besoin dans son langage métier.

L'utilisation d'un langage de modélisation capable d'élaborer un modèle, tel que nous l'avons décrit au chapitre précédent prend alors tout son sens.

L'objectif principal de cet ouvrage est d'offrir aux réalisateurs de bases de données relationnelles, une approche pragmatique du langage de modélisation proposé par la technique du Modèle Entité Association (M.E.A), reprise et développée dans la méthode Merise.

L'universalité des problématiques liées à ce type de conception montre l'intérêt de maîtriser un tel outil.

La méthode Merise a repris pour la partie concernant la modélisation de données la technique du MEA en l'intégrant au schéma global de l'approche à travers les différents niveaux d'abstraction. Cette technique est intégrée au plus haut niveau de la méthode, le niveau conceptuel, et donne lieu à la réalisation du Modèle Conceptuel des Données (MCD). A partir du MCD, on réalise le passage à la base de données selon un processus qui sera décrit au chapitre 2, paragraphe 4.

Pour ce qui concerne les traitements, la méthode Merise propose l'implantation au niveau conceptuel, de l'approche événementielle à travers, entre autres, les réseaux de Pétri. Ceci dépasse le cadre de ce livre et ne sera pas abordé dans cet ouvrage.

La notion de base de données a été introduite au milieu des années soixante dix, pour pallier les faiblesses techniques sur l'intégrité des informations contenues dans les fichiers de données utilisés à ce moment là pour stocker l'information.

Le paragraphe suivant compare les notions de bases de données et de fichiers, et leur intérêt respectif pour le développement des applications et des systèmes d'information.

3. Fichiers, bases de données, bases de données relationnelles

Nous avons vu que le problème du stockage de l'information est intimement lié à celui des fonctionnalités attendues par une application informatique.

Deux types de stockage sont à envisager :

Le stockage en mémoire primaire : la caractéristique principale de ce type de mémoire est la durée de vie des informations qui ne peut excéder celle de l'utilisation de l'application. Pratiquement, il s'agit d'informations stockées en mémoire vive de l'ordinateur dont la pérennité est égale au mieux à celle de la présence de l'application en mémoire centrale. Ce type de stockage n'est pas concerné par notre travail.

Le stockage en mémoire secondaire : Sa caractéristique principale réside dans la *persistance* des informations stockées. L'arrêt du fonctionnement du système ne provoque pas la perte des informations qu'il contient. Le support utilisé est physiquement étudié pour offrir cette caractéristique. Les principaux supports utilisés pour ce genre de mémoire sont les supports magnétiques (Disques dur) et les supports optiques (CD, DVD, etc.). Ce sont ces systèmes de mémorisation permanente qui nous intéressent ici.

3.1 Problématique aval : adressage logique et adressage physique

Une des propriétés les plus remarquables des supports de mémorisation persistants utilisés aujourd'hui, réside dans leur adressabilité. Tous ces supports offrent un mécanisme d'adressage de l'information fiable et efficace, complexe dans le détail de sa

mise en œuvre, mais simple sur son principe : chaque information possède une adresse sur le support. A partir de cette adresse, il est possible de récupérer l'information. Cette adresse est appelée adresse physique, car elle correspond à l'emplacement réel de l'information sur le support. Une fois l'information mémorisée, Le problème de sa récupération est évidemment le plus important.

Afin d'en simplifier l'utilisation, on associe à chaque adresse physique sur le support, une adresse logique plus simple à gérer. Vu de l'application, toutes les informations sont accessibles grâce à leur adresse logique.

Les développeurs se sont rapidement rendus compte de la nécessité d'automatiser la gestion de la correspondance : **adresse logique** \Leftrightarrow **adresse physique**.

Ce besoin, associé à la résolution de problèmes techniques plus spécifiquement liés aux supports, a conduit à la création des Systèmes de Gestion de Fichiers (SGF), fonctionnalité indispensable des systèmes d'exploitation.

Le SGF constitue l'outil privilégié de la gestion des accès aux supports persistants de stockage de l'information. Il offre une interface utilisateur standardisée, simple et conviviale afin de simplifier l'accès aux données.

Au sein des SGF, on distingue 3 niveaux sémantiques pour le stockage de l'information :

Le **champ** : unité élémentaire de stockage de l'information. L'information stockée est de type standard (numérique, chaîne de caractères, booléen) ;

L'**enregistrement** : Un enregistrement est composé de un ou plusieurs champs. C'est la plus petite unité d'information accessible en une seule opération d'entrée/sortie entre l'unité centrale et le support secondaire.

Le **fichier** : ensemble d'enregistrements. Quand une application désire accéder à certains enregistrements d'un fichier, le SGF ouvre un canal de communication entre l'unité centrale et le fichier, contenu sur le support de stockage physique. Le SGF réalise ensuite, à la demande, la conversion des adresses logiques en adresses physiques afin d'offrir les accès aux données désirées.

Le SGF est un outil permettant la standardisation des accès aux supports afin de simplifier le travail du programmeur. Le problème de la sémantique des informations n'est pas pris en compte au niveau des SGF. **En aucun cas**, il offre des moyens automatiques pour assurer la vérification de l'intégrité des informations stockées. Dans l'approche par les fichiers, cette tâche est laissée à la charge de l'application.

3.2 Problématique amont : l'intégrité des données

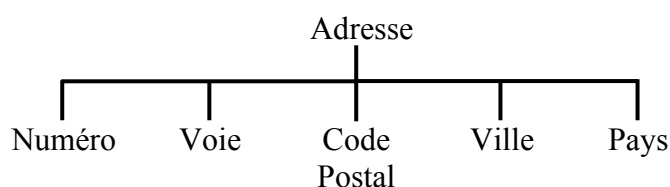
Le développement des capacités techniques des outils de mémorisation, associé aux impératifs économiques du développement d'applications informatiques, ont vite rendu indispensable le développement d'un outil pouvant assurer un certain niveau de fiabilité aux informations stockées dans les fichiers.

Or, le stockage de l'information est à envisager de deux points de vue: d'un côté le support physique et de l'autre les utilisateurs. Ceci conduit à envisager deux types d'outils : l'un prenant en charge l'interface avec le support physique et l'autre assurant la relation avec l'environnement utilisateur.

Le SGF a été associé au premier, et le Système de Gestion de Bases de Données (SGBD), correspond au deuxième. Le SGBD encapsule le SGF, c'est-à-dire qu'il en possède toutes les fonctions, et qu'il offre en plus des **moyens automatisés de contrôle de la cohérence des informations** stockées.

Pour assurer cette *intégrité des données* contenues dans les fichiers, **une hypothèse sur la structure de l'information** doit être envisagée. A chacune d'elle correspond un ensemble de règles à respecter pour assurer l'intégrité des données stockées.

Une des premières hypothèses faites pour contrôler la cohérence des informations a été l'hypothèse *hiérarchique* : selon cette approche l'information peut se décomposer en un ensemble d'informations pouvant elles-mêmes se décomposer, et ainsi de suite selon une approche composés/composants. On aboutit ainsi à une représentation conceptuelle arborescente. L'intérêt principal de cette approche est son apparente proximité avec ce que l'on rencontre dans la réalité. Par exemple, une adresse se compose d'un numéro, d'une voie, d'un code postal, d'une ville, et d'un pays, informations que l'on peut représenter hiérarchiquement de la façon suivante :



On a ainsi constitué des bases de données hiérarchiques associées à des SGBD offrant des outils adaptés à cette organisation, comme, par exemple, des fonctions pour le parcours d'arbres. L'étude des limites de cette approche dépasse le cadre de cet ouvrage, mais on peut dire que l'utilisation de données ainsi organisées est vite complexe et pose des problèmes d'efficacité particulièrement dans les parcours transversaux.

L'approche qui nous intéresse dans cet ouvrage repose sur l'hypothèse relationnelle. Celle-ci considère qu'il existe des relations entre les informations contenues dans un système. Parmi tous les types de relations possibles, celui qui nous intéresse est le fait que la connaissance d'une information en détermine une autre, une espèce de *causalité* entre informations élémentaires.

Dans l'exemple cité plus haut, on rencontre ce type de relation informationnelle entre le code postal et la ville. Il arrive que la relation inverse soit vraie aussi.

Un autre exemple, dans une entreprise, la relation qui peut exister entre une commande et un client : la connaissance de la commande détermine le client qui a commandé, mais, en général, dans ce cas, l'inverse n'est pas vrai. En effet un client peut passer plusieurs commandes mais une commande n'est passée que par un seul client, donc connaître le client ne suffit pas à déterminer la commande. La relation qui existe entre commande et client n'est pas symétrique.

Ces deux exemples montrent que l'approche par les relations entre les informations d'un système nécessite une analyse plus fine que dans le cas hiérarchique.

Dans le dernier exemple, la relation entre la commande et le client peut s'exprimer ainsi : *la connaissance de la commande détermine le client*. C'est ce type de relation que l'hypothèse relationnelle demande de mettre en évidence.

Cette présentation intuitive, et apparemment simpliste, de l'hypothèse relationnelle cache en réalité un fondement théorique mathématique extrêmement solide qui fait appel à la théorie des ensembles et des fonctions, ainsi qu'à l'algèbre relationnelle.

Au niveau du stockage de l'information sur le support, l'hypothèse relationnelle permet de définir des SGBD *Relationnels* (SGBD/R) qui proposent des fonctions de contrôle de l'intégrité des données associées à des fonctions d'accès aux données s'appuyant sur ces fondements mathématiques.

Afin de mieux distinguer les SGF et les SGBD Relationnels, les trois niveaux de stockage de l'information définis dans les SGF, ont été repris et renommés dans les SGBD/R afin d'éviter toute confusion entre fichiers et bases de données.

On distingue :

La **propriété** : c'est l'équivalent du champ dans les SGF, mais le contenu peut être associé à une ou plusieurs contraintes de validité. *Exemple* : Si le système correspond à une école primaire, l'information stockée dans la propriété *âge* de chaque élève pourra être contrôlée par rapport à une fourchette de valeurs comprises entre 6 et 12 ans. Ce contrôle peut être pris en charge par le SGBD/R.

Le **tuple** : c'est l'équivalent de l'enregistrement des SGF, mais il inclut en plus un certain nombre de contrôles possibles sur la validité et la cohérence des informations. Bien qu'impropre, le terme *enregistrement* est souvent utilisé.

La **table** : c'est l'équivalent du fichier dans les SGF. Outre la cohérence sémantique des informations qu'il contient, on peut lui associer un certain nombre de contrôles, par exemple sur les mises à jour des tuples de la table et des **tuples des tables auxquelles elle peut être liée**. Cette prise en charge d'un lien informationnel entre les tables par le système, est caractéristique et justifie la dénomination SGBD *Relationnel*.

Les SGBD/R (Système de Gestion de Bases de Données Relationnelles) offrent un ensemble d'outils de contrôle de la cohérence et de l'intégrité des informations contenues dans les bases de données en s'appuyant sur l'hypothèse relationnelle. Véritable Système d'exploitation des informations, tant du côté aval, à travers le SGF qu'il encapsule, que du côté amont à travers les outils qu'il offre aux utilisateurs pour un usage plus sûr et mieux contrôlé des informations. (cf. schéma 2)

Parmi les outils proposés aux utilisateurs pour la gestion des données, les SGBD/R offrent une interface d'accès aux données, basée sur le langage SQL (Search Query Language). Ce langage permet de s'affranchir des problèmes d'accès aux données en offrant une *expression non procédurale* des instructions sous la forme de phrases proches du langage courant. Il a été standardisé par l'ISO et offre des mises à jours régulières.

Le SGF offre une interface d'accès aux données afin de s'affranchir des contraintes liées à la programmation des accès. Le SGBD offre une interface d'accès sémantiquement contrôlée orientée métier. Quand l'hypothèse relationnelle est retenue, les mécanismes de contrôle mettent en jeu les résultats de l'algèbre relationnelle et de la théorie des ensembles.

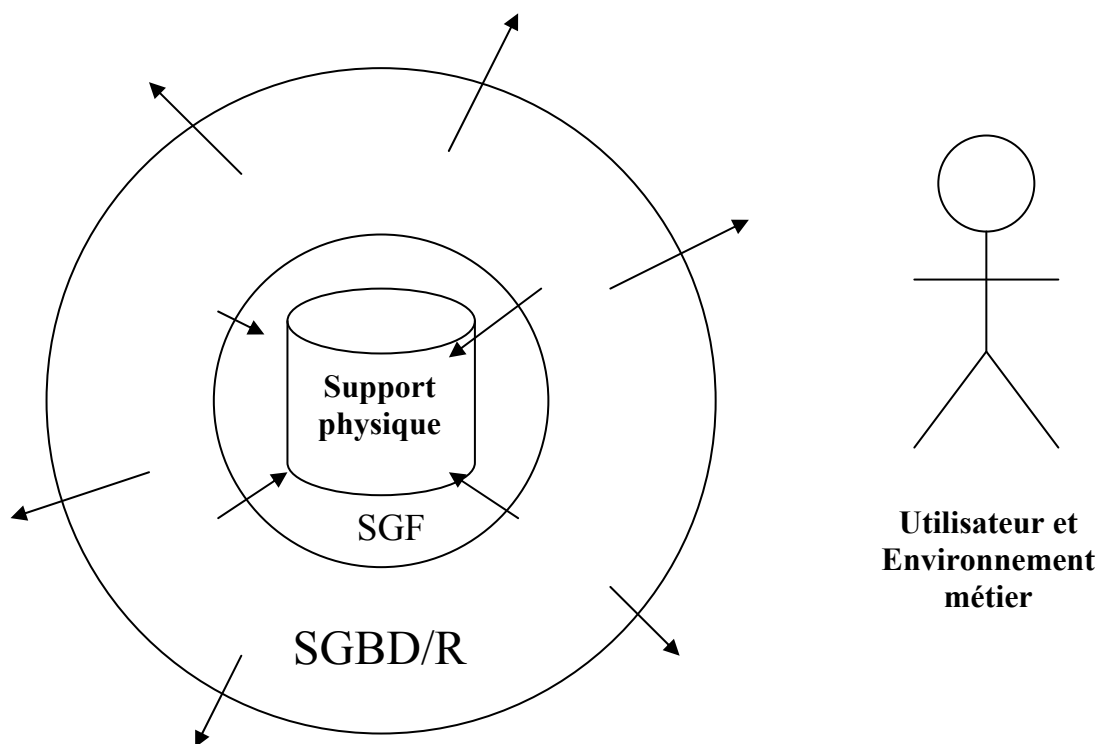


Schéma 2
L'encapsulation du SGF par le SGBD/R

Nous avons tenté de montrer dans ce paragraphe l'intérêt des bases de données relationnelles. C'est aujourd'hui un standard de fait.

L'efficacité dans leur mise en œuvre passe par l'utilisation d'une méthode de conception adaptée, permettant à l'utilisateur de concevoir un modèle compatible avec l'environnement relationnel envisagé. Comme nous l'avons montré au paragraphe 1, le langage de modélisation utilisé doit être choisi en fonction du langage utilisé par le maître d'œuvre. Ici, le SGBD étant Relationnel, le langage choisi sera celui du MEA, tel qu'il est proposé dans la méthode Merise.