

Description des fonctions d'un logiciel

Alain VAILLY

Département Informatique - UFR de Sciences et Techniques
Université de Nantes

I) Introduction générale

Deux objectifs ont été assignés à ce document, le premier étant de mieux faire comprendre ce qui se « cache » derrière la notion de **fonction** d'un logiciel (volet *méthodologie*) et le second de montrer comment rédiger un document décrivant ces fonctions à destination des utilisateurs (volet *communication*).

Destiné aux étudiants, ce document est construit à partir de textes et de schémas tirés d'études de cas « réels » (*realistic* serait plus juste). Afin de bien distinguer ce qui relève de la communication de ce qui est purement méthodologique, deux polices de caractères différentes sont utilisées. Dans le même but, nous rappelons à chaque paragraphe si le texte relève de la méthodologie ou de la communication.

Dans la mesure où, à chaque fois que cela sera nécessaire, des commentaires de méthode sont fournis, des extraits de documents « réels » donnés, une troisième « police » de caractères, en italique, est également employée. Elle est affectée aux textes relevant de la pédagogie.

I.1) Introduction (aspect méthodologie)

L'objectif de cette introduction est de situer la notion de fonction dans les différents aspects à prendre en compte lors de la définition de l'architecture d'un logiciel et d'esquisser ce qu'il faut faire pour les décrire.

Lorsque l'on définit un logiciel et, plus précisément, son plan, il y a trois points de vue à prendre en compte, celui des données, celui des fonctions et celui des comportements. L'étude selon le premier point de vue va nécessiter un travail de collecte des informations, de sélection de celles qui sont utiles et de structuration en ce qui deviendra une base de données. Prendre en compte le point de vue des fonctions, c'est étudier ce que veut l'utilisateur en terme de procédures, de données nécessaires à leur réalisation, c'est commencer à réfléchir aux écrans, états à produire. C'est aussi envisager le dialogue qui va s'établir entre le ou les utilisateurs finaux et le système. L'ordonnancement de ces fonctions, l'agencement de celles-ci les unes par rapport aux autres est du ressort du troisième point de vue.

Ces trois points de vue doivent être minutieusement coordonnés, confrontés de façon à produire un véritable effet stéréo. Le relief de l'application sera fourni par cette confrontation. L'épicentre de celle-ci se situe au niveau des fonctions qui mêlent données, traitements et dialogue, soit les trois points de vue. Les données nécessaires à la réalisation du traitement doivent faire partie de la structure de données. Le dialogue qui s'instaure entre le système et le client (le « fameux » utilisateur final) s'inscrira naturellement dans la dimension des comportements (même s'il ne s'agit que d'un aspect microscopique de cette dynamique). Le descriptif du traitement est par nature algorithmique (grossièrement algorithmique, d'un certain point de vue, mais algorithmique quand même).

Ce document donne des pistes pour mettre en évidence les fonctions. Il décrit ce qu'un architecte logiciel doit réunir comme informations sur ces fonctions, donne des exemples de celles-ci et esquisse le devenir de ces fonctions. Il est clairement destiné à un informaticien et/ou à un élève informaticien qui ambitionne de créer ses propres plans de logiciels. Il suppose de sa part un minimum de connaissances sur la spécification, sur la conception et même sur la programmation.

I.2) Introduction (aspect communication)

L'objectif de cette introduction est d'expliquer la notion de fonction à un public qui a priori n'est pas informaticien. Ce type de lecteur a besoin d'en comprendre le rôle et les objectifs, en termes « simples », non techniques. L'exemple donné ici est assez générique. Il peut être utilisé pour pratiquement toutes les introductions quel que soit le cas.

Qui dit gérer (un hôpital, un établissement bancaire, un aéro-club...) dit traitements. Il faut recueillir les informations auprès des utilisateurs, les transformer, en mémoriser certaines, produire des documents... bref, mettre en œuvre un ensemble de procédures qui, mises « bout à bout », constitueront l'outil de gestion de l'organisme.

Ces procédures doivent être minutieusement décortiquées, analysées, reconstituées pour pouvoir être reproduites mécaniquement. L'homme va, ici, céder sa place à la machine. Il convient que la copie soit aussi fidèle que l'original, faute de quoi non seulement il n'y aura pas amélioration des conditions de travail de l'utilisateur mais, en plus, on assistera à une détérioration de celles-ci.

Certaines procédures nécessitent une part importante d'automatisation ; d'autres sont presque exclusivement manuelles. Dans la plupart des cas, l'homme se chargera de tous les cas particuliers, du redressement des erreurs, la machine prenant en compte ce qui est routinier.

Il y a plusieurs avantages à ceci : outre le fait que l'informatisation, en réalisant la partie fastidieuse, laisse à l'homme la partie la plus intéressante de son travail, il y a création d'un « centre de renseignements », dans lequel sont décrits tous les traitements assurés par l'entreprise. On peut ainsi former (ou informer) d'autres personnes à ce qui est fait, sans perturber celles qui mettent en œuvre le traitement.

Ce document décrit chacune des fonctions que doit assurer le système. Il fait référence au dossier dans lequel est décrite la structuration des données, sans entrer dans le détail, le schéma conceptuel des données détaillé étant cependant fourni en annexe. Le comportement global du système est décrit dans un document spécial. Le lecteur peut ou non avoir pris connaissance de son contenu avant de lire celui-ci.

II) Mise en évidence des fonctions

Les fonctions correspondent aux services que rend le logiciel. Ce sont les traitements mis en œuvre. On peut distinguer trois grandes catégories :

- La première regroupe ce que demande l'utilisateur à l'architecte, plus ou moins précisément, au travers des entretiens. Il est fréquent qu'une liste soit établie par cet utilisateur et remise à l'informaticien. Voici, par exemple, celle fournie par le gestionnaire d'un aéroclub :

Notre interlocuteur a défini les objectifs que nous devons atteindre sous forme de phrases. Chacune des vingt-quatre procédures fonctionnelles est ainsi décrite :

PF 10 {Inscription d'un pilote}

Saisie des caractéristiques d'un nouveau pilote, ouverture d'un compte, versement initial, retrait de la cotisation calculée au prorata de l'année civile en cours.

PF 20 {Démission d'un pilote}

Suppression d'un pilote de l'ensemble des membres du club. Solder le compte, fermeture du compte si solde non débiteur, sinon fermeture différée jusqu'à la régularisation du compte.

PF 30 {Edition des noms d'un pilote avec ou sans caractéristiques}

Liste des noms de l'ensemble des pilotes du club. Chaque nom peut être accompagné des caractéristiques d'un pilote -informations relatives à un pilote-.

PF 40 {Edition des caractéristiques d'un pilote}

Edition individuelle des informations d'un pilote.

PF 50 {Edition des vols d'un pilote}

Liste de l'ensemble des informations des vols effectués par un pilote pour une période donnée.

PF 60 {Edition d'un relevé de compte d'un pilote}

Elaboration d'un relevé d'un compte présentant les débits et les crédits pour une période donnée -correspondant à une demande de l'opérateur.

PF 65 {Edition d'un relevé de compte pour l'ensemble des pilotes}

Effectue l'opération précédente pour l'ensemble des pilotes -cette opération peut être déclenchée manuellement ou automatiquement en fonction de la fixation d'une date périodique -cf paramétrage du système-.

PF 70 {Demande de position -débiteur ou créditeur- d'un pilote}

Interrogation du système pour connaître l'état du compte d'un pilote -commande interactive et pouvant être faite par programme-.

PF 80 {Débit sur le compte d'un pilote}

Opération de retrait d'une somme d'argent sur le compte d'un pilote. Chaque débit possède un libellé spécifiant l'opération -paiement d'un vol, cotisation du club, bar (boisson, repas, tabac, etc), accessoires boutique (pin's, casquette, blouson, serviettes, etc)- .

PF 90 {Crédit sur le compte d'un pilote}

Opération de versement d'une somme d'argent sur le compte d'un pilote. Chaque versement possède un libellé spécifiant le mode de l'opération -chèque, espèces, carte de crédit, virement, etc-.

PF 100 {Achat d'un avion}

L'achat d'un avion introduit un nouvel avion utilisable par les pilotes du club. La gestion du financement d'un avion n'est pas prise en compte. Par simplification, nous considérons que l'achat est effectué au comptant par le club. La table des tarifs est éventuellement mise à jour si l'avion correspond à une nouvelle catégorie.

PF110 {Vente d'un avion}

La vente d'un avion supprime un appareil utilisable par les pilotes du club. La table des tarifs est mise à jour si la catégorie correspondante à l'avion vendu n'a plus de représentant.

PF 120 {Perte d'un avion}

Suppression d'un avion de la liste des appareils disponibles. Pas de rentrée d'argent, si ce n'est éventuellement les primes d'assurances qui ne sont pas ici prises en comptes.

PF 130 {Edition des caractéristiques d'un avion}

Edition individuelle des informations relatives à un avion.

PF 140 {Edition du nombre d'heures de vol d'un avion}

Edition du nombre d'heures de vol d'un avion -interactive et par programme-.

PF 150 {Prise en compte d'un vol}

Enregistrement par le club d'un plan de vol. Un plan de vol implique un pilote, un avion et des informations sur le vol -heure et date de départ, durée estimée du vol, heure et date d'arrivée estimées, lieu de destination-.

PF 160 {Tarif horaire d'un avion}

L'ensemble des avions est partagé en plusieurs catégories en fonction de leur puissance et du nombre de passagers. A chaque catégorie correspond un tarif horaire. L'unité minimale est le quart d'heure. Le tarif horaire comprend le prix du carburant.

PF 170 {Edition des vols d'un avion}

Liste des informations des vols effectués par un avion pour une période donnée.

PF 180 {Edition bilan comptable du club}

Edition du bilan des mouvements effectués au sein du club. Permet de faire le point financier sur l'activité du club. Les seules opérations à considérer au niveau du club sont les achats et les ventes d'avions, les achats de fournitures et les ventes des heures de vol.

PF 190 {Modification des tarifs horaires}

Mise à jour de la table des tarifs horaires en fonction d'une augmentation annuelle des coûts, sur proposition de l'assemblée générale, ou sur un ajout ou sur une suppression d'une nouvelle catégorie d'avion.

PF 200 {Fixation des paramètres du système -date de consolidation générale, date d'édition des relevés des pilotes-}

La date de consolidation générale est fixée annuellement en prévision de l'assemblée générale pour permettre au trésorier -ou au comptable- d'établir le bilan financier du club et d'effectuer la relance auprès des membres du club pour le paiement de leur cotisation annuelle. La périodicité d'édition des relevés des pilotes est fixée par défaut à un mois.

PF 210 {Mise à jour d'informations erronées -fautes d'orthographe dans les noms, changements d'adresse, erreurs de saisies sur les vols, etc-}

Toutes les informations dépendantes d'une saisie manuelle peuvent faire l'objet d'une mise à jour. Les informations établies par le système ne sont pas modifiables -accès en lecture seulement-. Les modifications d'informations sensibles ne peuvent s'effectuer que par un opérateur autorisé -accès par mot de passe-. L'historique de ces modifications est conservé.

PF 220 {Démarrage et arrêt de l'application}

L'application est lancée à partir de la commande Gestion-Club. La sortie de l'application se fait par la commande Quitter. Une demande de sauvegarde des modifications en cours sera alors faite par le système.

PF 230 {Interface homme-machine}

L'interface de dialogue entre l'opérateur et l'application s'effectuera par l'intermédiaire de masques de saisie pré-définis, de menus. La frappe manuelle sera réduite au maximum. Le dialogue sera partitionné en distinguant les opérations de gestion générale au club, les opérations relatives aux pilotes, aux avions et aux vols. Les éditions pourront être faites soit interactivement à la demande sur écran, soit sur imprimante. La gestion des archives -fichiers annuels- n'est pas à prendre en compte.

Cette liste recense ce qui constituera le cœur de la future application. L'utilisateur compte fermement sur l'informaticien pour réaliser ces traitements. Ils sont, dans la plupart des cas, inscrits dans le cahier des charges remis au démarrage de l'étude.

- La deuxième catégorie regroupe des fonctions que l'utilisateur ne demandera presque jamais, mais qu'il serait fort surpris (c'est un euphémisme !) si le logiciel livré ne les proposait pas. Il compte sur l'informaticien pour les installer. On trouve dans cette catégorie des fonctions permettant de rectifier les erreurs survenues lors d'une saisie, qu'il s'agisse d'une entité ou d'une association. On y trouve aussi des fonctions d'édérations simplifiées (ie. sans mise en page soignée), d'archivage... bref, tout ce qui relève de la gestion de données numérisées, non spécifique d'une application.

- La troisième et dernière catégorie correspond à des traitements non demandés par l'utilisateur, mais proposées par l'informaticien. Il s'agit donc clairement d'extensions qu'il est simple de fournir et qui, mises à disposition de l'utilisateur, lui faciliteront la vie.

Comment mettre en évidence les fonctions ? Il y a plusieurs réponses à cette question et chaque analyste a ses propres recettes. En voici quelques-unes :

- Le point de départ est fourni par le client. Il nous procure, en effet, la liste de ce qu'il veut que le logiciel fasse.

- Un bon « truc » est de penser interface, menu et d'imaginer un menu avec des sous-menus et dans chacun une liste des fonctions disponibles, selon le modèle ci-après :

S/MENU 1	S/MENU 2	S/MENU 3	S/MENU 4	S/MENU 5
	Fonction A			
	<i>Fonction B</i>			
	<i>Fonction C</i>			
	Fonction D			
	Fonction E			

Il peut y avoir plusieurs menus, selon le niveau d'habilitation de l'utilisateur. Telle personne sera administrateur de la base, telle autre responsable de la bibliothèque, telle autre seulement lecteur... Chacune aura ses propres droits et donc sa propre liste de fonctions.

NB : la présentation faite ici en menus, sous-menus... n'est d'autre qu'un artifice. Elle ne préjuge en rien de l'implémentation ultérieure du logiciel.

- Un ordinateur travaille jour et nuit. Il est donc possible de programmer pour les nuits des tâches de fond, ie. ne réclamant pas la présence d'un utilisateur pour être exécutées, et de prévoir une mise à disposition de renseignements pour le tout début de la journée. Ainsi, par exemple, pourra-t-on mettre en place une fonction de scrutation des prêts pour signaler ceux qui arrivent à échéance le lendemain. Cette tâche sera exécutée chaque nuit, la liste des prêts se terminant enregistrée dans un fichier. Le matin suivant, le responsable des prêts pourra imprimer la liste ou la consulter sur son écran d'ordinateur.

III) Description des fonctions

La description d'une fonction doit être abordée selon trois angles, celui des informations nécessaires à la réalisation de la fonction, celui de l'algorithme à mettre en place pour atteindre le but fixé et celui du dialogue entre l'utilisateur et le système. Cette obligation d'une triple description est complètement indépendante de la méthode (Merise, UML...) que l'on suit. Nous reviendrons sur ce point plus tard dans le paragraphe.

L'ordre dans lequel ces trois études doivent être menées est quelconque. Elles doivent être conduites à leur terme toutes les trois, dans l'ordre que l'on veut. Nous préconisons toutefois de commencer par l'étude comportementale, puis de poursuivre en parallèle l'étude algorithmique et l'étude informationnelle. D'autres stratégies sont possibles.

III.1) Description des fonctions (aspect méthodologie)

Le lancement d'une fonction ne se fait pas sans tenir compte de l'environnement. Son résultat a également des répercussions sur celui-ci, en modifiant le contenu de la structure de données, ce qui va rendre possible l'exécution d'autres fonctions ou leur inhibition.

Dans un monde objet, basé sur l'exécution d'une tâche grâce à la collaboration de certaines classes pour atteindre l'objectif, le comportement de la fonction va concerner les messages échangés ainsi que le tuning de ces échanges.

Décrire le comportement d'une fonction revient donc à dire ce que fait la fonction lorsqu'elle est sollicitée, invoquée (non pas en termes algorithmiques, mais en termes de réactivité sur son environnement) et dire également quels sont les signaux qui l'active (la sollicite, l'invoque).

Dans un monde « classique », tel qu'il peut être décrit à l'aide de notations « à la Merise », la fonction fait partie d'un ensemble de tâches qui sont exécutées les unes après les autres, selon une logique exprimée par un modèle conceptuel de traitements.

Décrire le comportement d'une fonction revient alors à dire quelles sont les fonctions qui l'ont précédée, quels sont les événements qui sont survenus et qui ont rendu possible son exécution, quels sont ceux qui en résultent et quelle(s) fonction(s) devien(nen)t déclenchable(s) une fois terminée.

III.2) Description des fonctions (aspect communication)

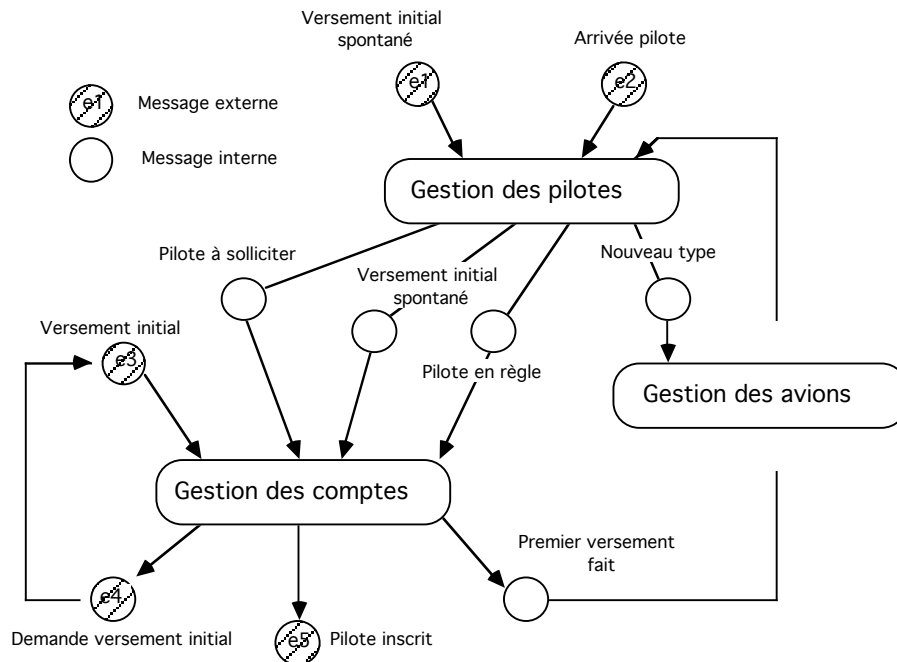
Cette partie illustre comment, sur deux cas concrets –celui d'un aéroclub et celui d'une université–, l'on peut décrire le comportement d'une fonction. Ces deux cas sont étudiés, chacun, dans un contexte différent, plutôt objet pour l'aéroclub, plutôt classique pour l'université.

Le premier exemple est extrait de la spécification d'un logiciel de gestion d'un aéroclub. Il concerne plus particulièrement la fonction d'inscription d'un nouveau pilote.

3-1-1) Inscription d'un nouveau pilote

a) point de vue comportemental

Le comportement du système que nous voulons construire peut être décrit sous la forme de réactions à des messages externes :



Trois “concepts” sont concernés, les pilotes, les avions et les comptes.

Le concept PILOTES réagit comme suit, à l'arrivée des différents messages :

{Arrivée d'un pilote}

ajout d'un pilote
état-pilote := “A-solliciter”
→ Pilote à solliciter
si nouveau type d'avion
→ Nouveau type

{Arrivée d'un pilote + Versement initial spontané}

ajout d'un pilote
état-pilote := “En-règle”
→ Pilote en règle
→ Versement initial spontané
si nouveau type d'avion
→ Nouveau type

{Premier versement fait}

état-pilote := “En-règle”

Le concept COMPTES, quant à lui, adopte le comportement ci-dessous décrit :

{Pilote à solliciter}

ajout d'un compte pour le pilote

ajout des mouvements (double écriture)
→ Demande de versement initial
{Pilote en règle + Versement initial spontané}
ajout d'un compte pour le pilote
ajout des mouvements (double écriture)
→ Pilote inscrit
{Versement initial}
ajout des mouvements (double écriture)
→ Pilote inscrit
→ Premier versement fait

Le concept AVIONS, enfin, va traiter le message "nouveau type" comme suit :

{Nouveau type}
ajout d'un nouveau type

NB : le langage que nous avons utilisé ci-dessus est très simplement basé sur les conventions d'écriture suivantes :

{xxxxx} arrivée du message xxxxx
→ xxxxx émission du message xxxxx

b) point de vue fonctionnel

On peut formaliser la procédure de la façon suivante :

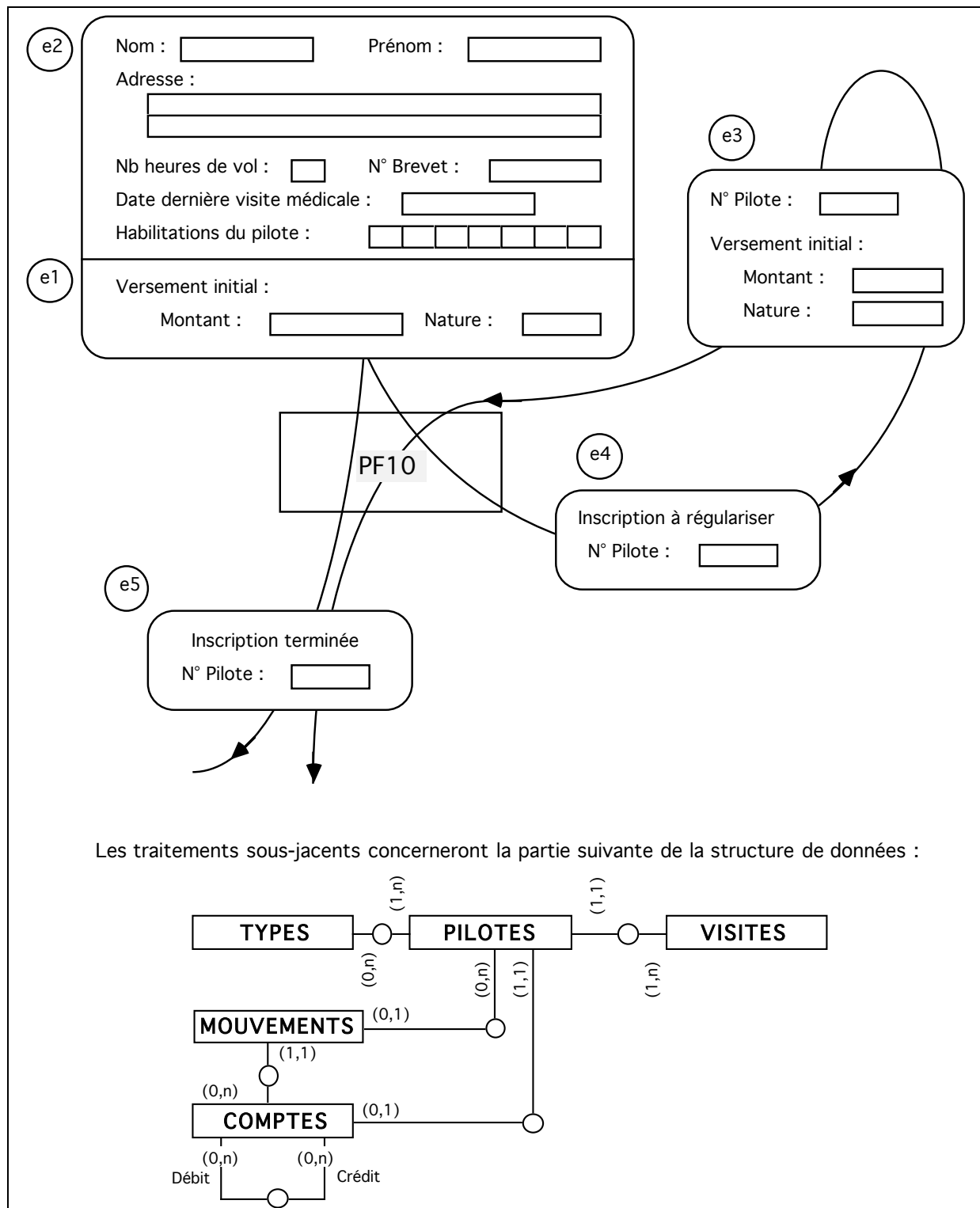
- Saisie du contenu message e2
- Ajout d'une occurrence de PILOTES
- Liaison de celle-ci aux habilitations (par PEUT-EFFECTUER, qui relie PILOTES à TYPES), avec création de nouveaux types s'ils n'existent pas encore
- Ajout d'une occurrence de VISITES
- Liaison de celle-ci à l'occurrence de PILOTES concernée

cette séquence réglant les problèmes administratifs. Il faut ensuite s'occuper de l'approvisionnement du compte du pilote :

- Saisie contenu message e1 ou e3
- Accès à l'occurrence de PILOTES concernée
- Création d'une occurrence de COMPTES (le compte du pilote)
- Enregistrement du versement initial
 - + Création d'une occurrence de MOUVEMENTS
(type := crédit ; compte du pilote)
 - + Liaison de celle-ci au compte correspondant et au pilote
 - + Création d'une occurrence de MOUVEMENTS
(type := débit ; compte "PILOTES" du club)
 - + Liaison de celle-ci au compte correspondant
- Enregistrement de la cotisation
 - + Création d'une occurrence de MOUVEMENTS
(type := débit ; compte du pilote)
 - + Liaison de celle-ci au compte correspondant et au pilote
 - + Création d'une occurrence de MOUVEMENTS
(type := crédit ; compte "PILOTES" du club)
 - + Liaison de celles-ci au compte correspondant

c) point de vue informationnel

L'utilisateur sera "confronté" à la série de messages suivants :



Le second exemple est tiré de la spécification d'un logiciel de gestion d'une université et concerne plus précisément les inscriptions.

5.4) : édition des listes

a) point de vue comportemental

La production des listes est déclenchée par le système, d'après le (pseudo) dialogue utilisateur ci-dessous :

PF5

↓

e1

Convocations au marché

Etudiants non inscrits à une filière ou inscrits à plusieurs

N°	Nom et prénom

Etudiants non inscrits à une UV ou inscrits à plus de 4

N°	Nom et prénom

Liste des UV ayant moins de 10 étudiants inscrits

N°	N°								

b) point de vue fonctionnel

La procédure qui permet de produire ces listes est la suivante :

Début

Affichage début message e1

Pour tous les étudiants faire

Selon cas de l'étudiant

pas inscrit à une filière : Affichage étudiant

inscrit à plus de 4 filières : Affichage étudiant

Finselon

Finpour

Pour tous les étudiants faire

Selon cas de l'étudiant

pas inscrit à une UV : Affichage étudiant

inscrit à plus de 4 filières : Affichage étudiant

Finselon

Finpour

Pour toutes les UV faire

Si moins de 10 étudiants inscrits alors

Affichage UV

Finsi

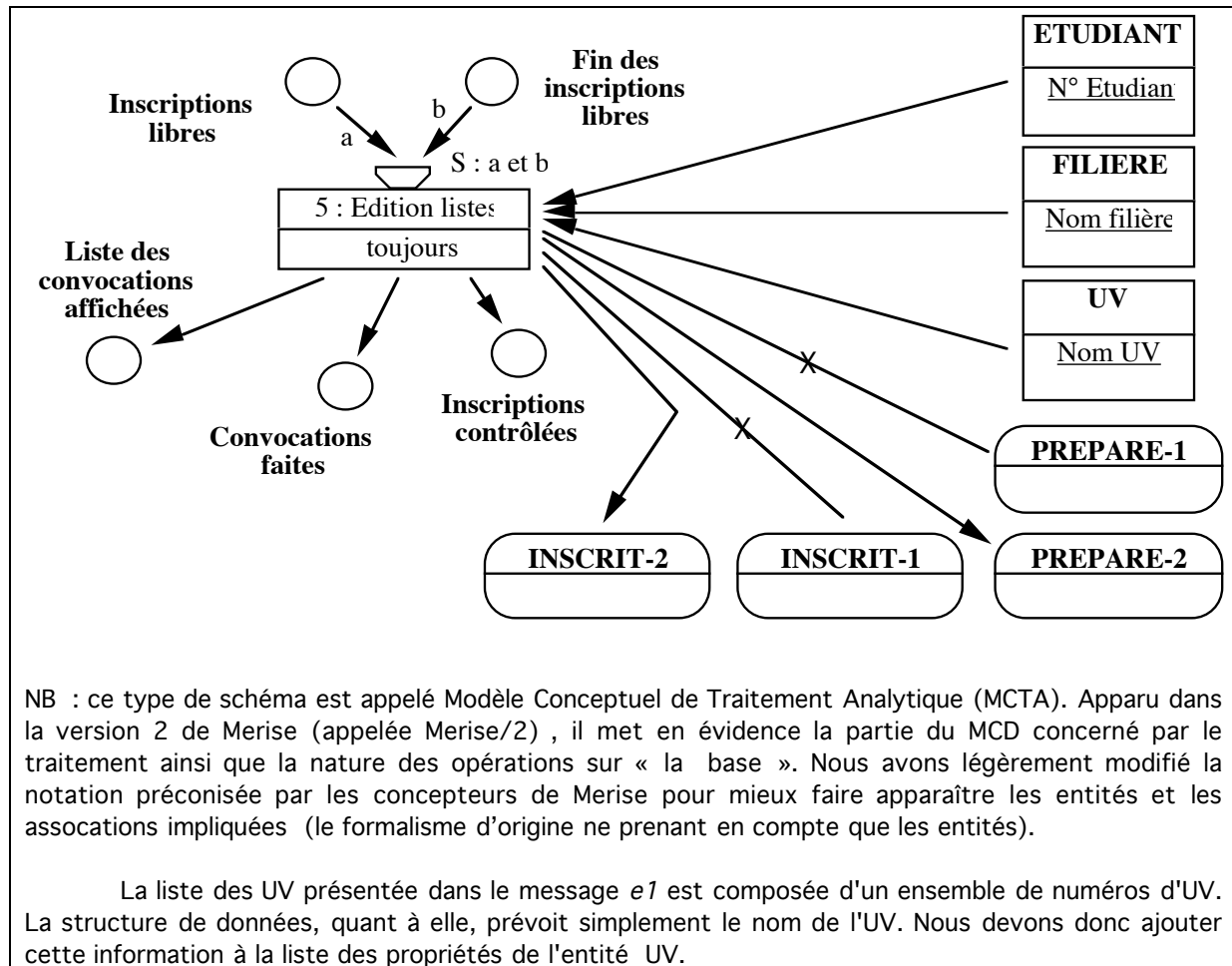
Finpour

Affichage fin message e1

Fin

c) point de vue informationnel

L'impact de ce traitement sur les données peut être résumé ainsi :



IV) Conclusion

Deux objectifs avaient été assignés à ce document. Le premier était de mieux faire comprendre aux étudiants ce qui se « cachait » derrière la notion de fonction. Pour répondre à cette question, nous avons mis en avant plusieurs recettes, montré au travers d'exemples ce que pouvaient donner celles-ci.

Le second était de montrer comment rédiger un document décrivant ces mêmes fonctions. Deux exemples, tirés chacun d'un cas différent, ont été fournis.

Nous pensons donc avoir atteint les buts fixés et nous nous tenons à la disposition des lecteurs pour toute explication complémentaire qu'ils pourraient souhaiter avoir.

IV.1) Conclusion (aspect méthodologique)

Les fonctions sont, peut-être, la partie la plus sensible d'un logiciel. C'est ce que voit le client pour qui nous travaillons, c'est sa trousse à outils avec laquelle il va résoudre les problèmes qui se posent à lui dans le cadre de son travail.

Nous ne pouvons donc pas nous permettre de nous « rater » dans la conception de cette partie. Pas assez de fonctions et le client ne pourra pas régler tous ses problèmes. Trop de fonctions est l'assurance d'un gaspillage de temps (pour l'informaticien qui en a trop fait) et de moyens (que le client n'utilisera pas).

Chaque fonction doit correspondre exactement à ce qu'attend le client, ni plus ni moins. Elle doit être l'objet de tous nos soins et être étudiée sous trois angles

complémentaires. Le premier est celui des données nécessaires à sa réalisation. C'est le point de vue informationnel. Le deuxième est celui du code à exécuter pour effectuer le travail attendu. C'est le point de vue fonctionnel. Le troisième et dernier envisage la fonction dans son environnement logiciel et humain. C'est le point de vue comportemental.

Les trois aspects sont nécessaires et aussi importants les uns que les autres.

IV.2) Conclusion (aspect communication)

L'objectif de cette conclusion est de resituer le travail dans le processus de développement, de rappeler de quoi parlait le document et d'esquisser quelques pistes possibles d'améliorations éventuelles. L'exemple fourni ici est extrait de l'étude réalisée pour l'automatisation d'un aéroclub. Il est certes spécifique, mais sa généralisation à d'autres cas est tout à fait envisageable.

L'outil que nous venons de définir est doté d'un certain nombre de fonctionnalités. Certaines, la plupart, répondent à des besoins explicitement formulés par l'utilisateur. D'autres vont permettre d'assurer l'entretien de la base. Le client ne les a pas mentionnées. Il n'envisage cependant pas qu'elles ne puissent exister. A quoi servirait, par exemple, un système qui ne pourrait être initialisé, ou nettoyé ?

Ces procédures fonctionnelles sont minutieusement décrites dans ce document, chacune étant étudiée d'un triple point de vue : comportemental (comment réagit le système ?), fonctionnel (que fait le système ?) et informationnel (de quelles données le système a-t-il besoin ?).

L'ensemble constitue une première réponse au "contrat" que nous avons passé avec l'Aéroclub du Castellet. Cette proposition peut, sans aucun doute, être affinée, améliorée, soit parce que l'utilisateur, à l'usage, aura de nouveaux besoins, soit parce que, durant la conception, quelques extensions possibles sont apparues.

Parmi celles-ci, on peut mentionner la surveillance "rapprochée" des comptes des pilotes (elle ne peut se faire actuellement qu'au moment des relevés de comptes), ainsi que le traitement des visites médicales et contrôles techniques subis par pilotes et avions (la mise à jour passant par les procédures de mise-à-jour globales). Il y en a probablement d'autres.

Cela pourrait faire l'objet d'un nouveau contrat.