

Construction d'un vocabulaire informatique

par Patrick Paumard

DÉFINIR LES MOTS.

Le vocabulaire de la langue française comprend neuf natures de mots, les plus courants sont les noms, les verbes, les adjectifs, les articles et les adverbes. Le nom permet de désigner un objet et le verbe de définir ou d'agir sur cet objet (état ou action).

L'ensemble des mots dans une construction de phrases est régie par les règles de la syntaxe grammaticale.

Dans les domaines techniques, seul le « nom » ou mot désignant l'objet est utilisé. La construction des mots est importante et doit suivre des règles strictes et cohérentes.

Ainsi en quincaillerie une vis est définie par différentes propriétés qui sont ordonnées, avec entre autres, sa destination (bois, tôle, acier), sa forme de tête, son diamètre, sa longueur, son pas, etc. Dans un domaine d'utilisation défini, les informations inutiles sont supprimées, et une vis normalisée est alors nommée uniquement par son diamètre et sa longueur, par exemple « Ø 8 × 125 ».

En ayant assimilé les règles simples de constructions de ce vocabulaire, très rapidement nous savons déchiffrer les mots et nous les comprenons.

En revanche, des constructeurs utilisent parfois des termes techniques mal construits, alors la nomenclature donne une liste de termes ésotériques, et la compréhension est impossible.

En informatique, une donnée désigne un objet qui peut être suivant l'application un montant, une date, un dessin ou un des ingrédients d'une recette de cuisine. La donnée, ou variable en APL, correspond au « nom de l'objet » de la vie courante.

Le traitement renseigne ou modifie l'état d'une donnée et a un pouvoir d'action sur cette donnée, par exemple « déplacer » , « multiplier par » , « imprimer » , « effacer » , « comparer » , Le traitement, ou fonction en APL, correspond au « verbe » de notre langue.

Les variables correspondent aux noms et les fonctions aux verbes.

Le vocabulaire, que nous utiliserons en APL, ne contient que deux natures de mots les noms et les verbes, la grammaire, extrêmement simple, nous obligera seulement à respecter les règles de positions des mots définis par le langage APL.

CRÉER UN VOCABULAIRE EXPLICITE.

En APL, les noms des variables ou des fonctions ne sont pratiquement pas limités en taille, jusqu'à 100 caractères en APL2000 et bien plus en DIALOG APL. Ceci à l'avantage de pouvoir donner à chaque mot un nom explicite, de telle sorte que le nom donné à un objet soit sans ambiguïté, et que le but du traitement soit implicitement compréhensible.

Exemple d'objets ou variables:

« Le_fauteuil_gris_du_bureau_de_ma_maison »
« Le_taux_d_interet_utilise_dans_le_calcul_des_amortissements »
« La_duree_du_remboursement_utilise_dans_le_calcul_des_amortissements »
 « Le_cote_oppose_a_l_angle_droit_dans_le_triangle_rectangle »
 « L_hypotenuse »

Les objets listés ci-dessus sont parfaitement définis, et aucun doute ne subsiste quant à leurs définitions. Dans le cas du fauteuil, nous supposons qu'il est unique, dans le cas où plusieurs fauteuils identiques existeraient et que nous ayons besoin de les distinguer, nous serions alors obligés d'utiliser un indexage.

Exemple de verbes ou fonctions:

« Pose_un_coussin_sur_un_fauteuil »
« Calcule_le_tableau_des_amortissements »
« Affiche_les_resultats_du_calcul_du_tableau_des_amortissements »
« Traite_graphiquement_les_actions_selectionnees »

Il est inutile d'expliquer les buts des fonctions présentées, à moins d'accepter la redondance du texte, les explications des fonctions étant identiques à leurs noms respectifs.

Notons l'absence de signes diacritiques et l'utilisation d'un signe particulier pour assurer la séparation des différents éléments composant le mot ou le verbe.

STRUCTURER LES MOTS.

Mais les noms composés ainsi ont un défaut majeur, lié à notre mode de lecture, nous lisons de gauche à droite. Ainsi les deux mots « antiroulis » et « roulis » ne seront liés que dans une liste par thème ou encyclopédique, et dans un dictionnaire, il n'y aura pas de concordance.

Les variables ou les fonctions traitant les mêmes données d'affichage et de calcul ne seront pas facilement rassemblées, et ce, quels que soient les critères de tri, ordre alphabétique sur les mots entiers ou uniquement sur les premiers arguments.

Sur le plan fonctionnel, contrairement à une liste traitant toutes les fonctions où apparaît la notion d'amortissement, la liste partielle des fonctions d'affichage n'a que peu d'intérêt.

En choisissant des noms semblables pour les fonctions, mais ordonnées sur l'importance décroissante des arguments, nous aurons naturellement des listes fonctionnellement cohérentes.

Les variables deviennent :

« Amortissement_duree »
« Amortissement_taux »
« Amortissement_annuite »
« ... »

Les fonctions quant à elles deviennent :

« Amortissement_tableau_calcule »
« Amortissement_tableau_resultat_affiche »
« Amortissement_tableau_resultat_imprime »
« Amortissement_tableau_resultat_..... »
« ... »

En listant les fonctions « Amortissement », nous obtenons cette fois-ci un lexique complet de toutes les fonctions gérant les tableaux d'amortissements.

Nous pouvons remarquer l'aspect modulaire du lexique. Pour modifier l'impression du tableau des résultats, nous irons naturellement modifier le module qui imprime, donc la fonction « Amortissement_tableau_resultat_imprime »

UTILISER DES PRÉFIXES ET DES SUFFIXES.

Dans les exemples ci-dessus, certaines variables et fonctions appartiennent à un ensemble plus vaste de variables et fonctions financières. Ainsi, il est judicieux de les préfixer par le nom de leur domaine d'application.

Exemple :

« Finance_amortissement_duree »
« Finance_amortissement_tableau_calcul »
« Finance_actions_monetaire_liste »
« Finance_actions_selectionne_traite_graphiquement »

Le nom composé de la dernière des fonctions listées ci-dessus utilise le mot « graphique ». Les graphiques sont gérés par une série de fonctions utilitaires, il sera alors facile, avec l'utilisation d'un préfixe tel que « Graphique » de les retrouver et de comprendre leurs enchaînements.

De même certains traitements obéissent à une cinématique stricte (début, répétition et fin), et certains objets sont décomposés en plusieurs parties distinctes. Par exemple une page est souvent

découpée en trois parties, qui sont l'entête, le pied et le corps. Ces trois parties et l'ensemble nous donneront les noms suivants :

« Préfixe_objet_page_entête »

« Préfixe_objet_page_pied »

« Préfixe_objet_page_corps »

« Préfixe_objet_page »

Dans certains cas, en fonction de la fréquence de l'usage d'un des mots, nous pourrions privilégier le corps de la page en prenant le mot sans le suffixe, soit « Préfixe_objet_page » et en qualifiant alors obligatoirement l'ensemble tel que « Préfixe_objet_page_tout ».

UTILISER LES ABRÉVIATIONS ET LES SIGLES.

Le vocabulaire de la vie courante fourmille d'abréviations comme « ampli » ou « moto », ainsi que de sigles comme « RER », « EDF » ou d'acronyme tel que « laser ».

Dans les domaines des sciences et technologies, des mots nouveaux sont inventés pour résumer les définitions des découvertes ou améliorations, par exemple « cédérom »

Plus le nom de la variable ou de la fonction est précis, plus ce nom devient long, un peu comme certains mots de la langue de Goethe ou dans le vocabulaire utilisé en chimie.

Ceci a un inconvénient majeur, la longueur du texte augmente le temps de lecture, donc la facilité de lecture du texte diminue, et la compréhension diminue également. Ainsi la précision nuit à la compréhension.

qualité de l'information ← précision → fatigue → incompréhension

En biochimie, les documents ou les articles de presse font souvent appel à des sigles, et dans les textes, les auteurs utilisent plus facilement les abréviations « ADN » et « ARN » que les termes « Acide désoxyribonucléique » et « Acide ribonucléique ». Ces sigles ont l'avantage, en permettant une lecture plus rapide, de laisser dans le même laps de temps, plus de temps au lecteur pour assimiler la phrase, et par la suite de la comprendre.

Dans la plupart des domaines techniques, nous utilisons les sigles.

En automobile, les « AOA » ou « RFE » sont dans tous les manuels professionnels, ils signifient respectivement « Avance Ouverture Admission » et « Retard Fermeture Echappement » en parlant du décalage angulaire entre le début du mouvement des soupapes d'admissions ou d'échappement, et le « PMH » ou « Point Mort Haut » qui lui enfin définit la position la plus haute du piston dans le cylindre.

En informatique, le mot « DOS » est plus employé que la suite de mots « Disk Operating System », et si la signification exacte est inconnue pour beaucoup d'utilisateurs de micro-ordinateur, la plupart connaissent ce mot, et en appréhendent plus ou moins le sens.

L'utilisation des abréviations et sigles permet d'augmenter la compréhension des textes sans perdre en précision.

Ainsi les variables et fonctions financières commenceront par l'abréviation « Fin » et les fonctions utilitaires de gestion des graphiques pourront être préfixés par une des abréviations « Graph » ou « Grf ». La seconde sera suffisante, car ce raccourci ne posera pas de problèmes de confusion, hormis peut être, avec le mot « Girafe ».

Exemple :

« Fin_amort_duree »

« Fin_actions_select_trait_graph »

« Grf_trait_init »

« Grf_symbol_convert ».

Nous pouvons remarquer que le mot « fin » opposé au mot « début » ne pourra être qu'un suffixe de verbe tel que « ...trait_fin », ou un qualificatif d'objet tel que « ...periode_fin ».

Pour mémoire, les abréviations se font de plusieurs manières, les plus usitées sont la troncation et la suppression des voyelles. Nous pouvons également utiliser les abréviations normalisées, mais attention « M » signifie « monsieur » et « m » l'unité de mesure « mètre ».

AMÉLIORER LA COMPRÉHENSION.

Nous pouvons encore améliorer la lisibilité et ainsi faciliter la compréhension, en respectant quelques règles simples pour la construction des noms de variables et fonctions.

Utilisons un vocabulaire simple, précis et juste.

La facilité de lecture d'un texte dépend bien sûr de la qualité du texte, du style de l'auteur, et de la présentation (corps des caractères, couleurs, colonne, etc.), mais aussi, et surtout dans notre domaine, de notre connaissance du vocabulaire utilisé dans ce texte.

La nécessité que nous aurions de consulter un dictionnaire à chacun des mots rencontrés ne rendrai que plus ardu la compréhension globale, puisque dans un premier temps l'effort va porter sur l'apprentissage de chacun des termes, leurs assimilations, et enfin la compréhension globale.

En pratique, nous utilisons un dictionnaire essentiellement pour rechercher les définitions des mots inconnus, si nous lisons « anachorète » nous consulterons peut-être le dictionnaire alors que nous n'irons pas chercher les définitions de « moine » et de « solitaire ». A l'instar, un mot comme « anticonstitutionnellement » ne nécessitera pas non plus de recherches, car nous savons le décomposer et sa compréhension est évidente.

Utilisons les majuscules pour séparer les différents composants d'un mot.

En effet les majuscules sont préférables aux tirets car elles permettent la diminution du nombre de caractères (une majuscule égale un tiret plus un caractère). Le tiret ne sera conservé que dans certains cas particuliers, notamment pour souligner visuellement les éléments d'un sous-ensemble, un peu comme l'utilisation de la ponctuation dans la phrase. Et si dans un premier temps, votre œil discerne mal les ruptures de ces constructions, très rapidement vous apprécierez l'avantage de ce procédé.

Exemple : EtSiDansUnPremierTempsVotreŒilDiscerneMalLesRupturesDeCes...

Évitons les mots inutiles ou redondants dans les constructions des noms.

Les mots « sélect » et « trait » sont superflus, en effet nous ne présenterons graphiquement que les objets qui sont sélectionnés, et les fonctions, en général, traitent quelque chose.

Les noms des fonctions deviennent « GrfInit » et « FinActionsGraph ».

Écrivons différemment les variables des fonctions.

Dans la langue allemande, au contraire des autres mots, tous les noms débutent par une majuscule (en français, seulement les noms propres).

En informatique, à l'inverse, nous commencerons les noms par des minuscules, et les verbes par des majuscules. Ainsi, les préfixes identiques pour les fonctions et les variables, seront différenciés par la case de la première lettre.

Le choix, minuscule / majuscule s'explique pour les fonctions et variables, car si la donnée est plus importante que le traitement, c'est ce dernier qui est primordial pour l'informaticien. Nous pouvons ainsi, aisément distinguer les variables des fonctions.

Les variables deviennent

« grfTitre »

« grfAxeX_Titre »

« grfAxeX_Unite »

« grfAxeY_BorneSup»

Les fonctions quant à elles deviennent

« FinActionsGraph »

« GrfInit »

« GrfSymbolConvert »

Notons pour mémoire, que dans les anciennes versions d'APL gros système, la différenciation pouvait être faite par le soulignement (seuls les caractères majuscules existaient).

Utilisons le polymorphisme pour diminuer le nombre de fonctions.

Le polymorphisme, en langage orienté objet, est la possibilité d'associer à plusieurs objets une seule et même méthode. En fait, c'est la possibilité d'avoir plusieurs fois le même nom de méthode, pour désigner des protocoles différents, en quelque sorte pour un seul nom, il y a plusieurs formes d'actions.

En langage courant, prenons le verbe « ouvrir », bien que l'action soit totalement différente si elle s'applique à une fenêtre, à une porte d'entrée (avec serrure), à une boîte de conserve, à un pack de soda ou à une bouteille de vin nous utilisons le même verbe (même si certains esprits chagrins disent que nous pouvons utiliser « déverrouiller » pour la porte et « déboucher » pour la bouteille).

Les actions sont différentes et les outils employés sont dissemblables, aucun pour la fenêtre, une clé, un ouvre-boîtes, etc., il n'y a bien qu'un seul verbe, qui prend des formes différentes, pour

nommer pour diverses actions.

En APL nous pouvons utiliser le polymorphisme dans différents cas, prenons la fonction « SStoMAT » elle traite indifféremment un vecteur de données alphabétiques ou numériques. Nous avons une seule fonction pour traiter des données de genres différents.

Un cas intéressant d'utilisation est celui de l'interrogation / modification.

Exemple :

l'interrogation du paramètre « Parm » dans la section « Section » du fichier WIN.INI

```
WindowsWinIniValue 'Section' 'Parm'
```

et la modification de ce même paramètre

```
'NouvelleValeur' WindowsWinIniValue 'Section' 'Parm'
```

Vérifions la constance des abréviations et la cohérence des structures.

Il est important de bien choisir une abréviation et de n'utiliser que celle-ci, prenons le mot « date », les abréviations possibles sont « dt », « dat » et « dte », quelque soit celle que nous choisissons, nous ne devons pas employée les autres.

De même la structure de construction des noms d'un groupe de variables ou de fonctions doit être constante

Exemple :

Lister un répertoire du DOS → DosDIR ou DosDirListe

Créer un répertoire du DOS → DosMD ou DosDirCree

Supprimer un répertoire du DOS → DosRD ou DosDirSupp

Dans les cas présentés ci-dessus, chacune des options choisies ayant son intérêt, il faut choisir comme termes : soit des noms identiques aux commandes du DOS ; soit des noms explicitant les commandes du DOS.

Nota : « DIR » pour « directory » en anglais et « répertoire » en français.

De même

Ouverture d'un fichier DOS → DosFileOpen ou DosFiOuvre

Fermeture d'un fichier DOS → DosFileClose ou DosFiFerme

Nous choisirons comme racine « Fi » ou « File » , et la liste anglaise des instructions ou leurs traductions avec les verbes français correspondants, toutes les autres fonctions devront être construites avec les mêmes composants.

CONCLUSION.

En respectant ces quelques recettes de construction d'un vocabulaire clair et précis, le code informatique devient plus aisé à lire, ainsi l'attention du lecteur se portera plus facilement sur les complexités réelles des programmes, et la compréhension des algorithmes, des enchaînements, etc., en sera améliorée.