

Les 8 défauts du procédural

Le procédural est une vieille technique de programmation mais c'est toujours la seule utilisée aujourd'hui. C'est grâce à elle que notre société actuelle existe et a pu se développer. Sans Le procédural, l'Intelligence Artificielle elle-même ne peut exister car il faut bien la programmer elle-même... Pour le moment, le procédural – avec l'algorithmique - sont incontournables pour l'écriture des logiciels simples, la programmation de calculs ou d'actes réflexes. En somme, pour les tâches ne réclamant pas de raisonnement.

Maintenant, passons en revue les défauts du procédural. La liste qui suit, vous ne la trouverez nulle part... Il faut être non informaticien pour oser la commettre ! J'ai mis l'Intelligence Artificielle raisonnante en parallèle (la Maïeutique) pour que vous puissiez mieux mesurer à quel point ses avantages sont énormes. Ils sont au nombre de huit :

1. *Le procédural nécessite la présence d'un intermédiaire*
2. *il représente la connaissance de façon illisible pour le commun des mortels*
3. *il faut coder tous les cas qui se présenteront*
4. *le programmeur fait un boulot de dingue ...et il en est fier*
5. *pendant tout le développement, l'expert ne voit pas son programme !*
6. *on ne touche plus au programme une fois écrit*
7. *impossible de modifier un programme en cours d'utilisation*
8. *impossible d'aborder certains domaines d'applications*

● 1^{er} défaut : *Le procédural nécessite la présence d'un intermédiaire*

Entre l'expert qui détient la connaissance à mettre dans le programme et les utilisateurs qui utiliseront le programme, il y a un intermédiaire : le programmeur. C'est lui qui est chargé d'écrire le programme. C'est lui - ou un autre - qui sera par la suite chargé de le faire évoluer. Comme il faut tout lui expliquer, c'est long et fastidieux. De toute façon, il ne peut pas comprendre une expertise qui a demandé des années à être acquise par l'expert. ça ne l'empêche pas, par un amour-propre compréhensible, de prétendre le contraire, ce qui a pour effet d'envenimer fréquemment les relations informatique-services utilisateurs...

Avec la Maïeutique, c'est l'expert qui écrit l'application et la fait lui-même évoluer, en collaboration avec les utilisateurs. Une fois terminée, il la remet au service informatique pour intégration dans le système d'informations. Le poids de l'erreur dans les logiciels repose désormais sur les experts (qui s'en accommodent très bien...). Les informaticiens sont libérés des tensions avec les services utilisateurs. Ils deviennent des collaborateurs « normaux » parlant le langage de tous.

● 2^{ème} défaut : *Le procédural représente la connaissance de façon illisible pour le commun des mortels*

Un programme, ce sont des pages de « code » cryptées dans un langage hermétique qu'on ne peut comprendre qu'après avoir suivi une longue formation. Même le programmeur ne s'y retrouve pas quand il remet le nez dans un programme qu'il a écrit il y a plusieurs mois... Et je ne parle pas du malheureux qui doit se replonger dans un programme écrit par un autre !

La Maïeutique présente, elle, un savoir-faire ET une connaissance humaine en langage courant, qu'elle sait « faire tourner » comme du procédural bien écrit. Avec elle, le programme se réduit à une petite base de données toute bête : une « base de connaissances ». C'est quelque chose d'infiniment plus concis et clair que du procédural.

● 3^{ème} défaut du procédural : *il faut coder tous les cas qui se présenteront*

Que fait l'expert quand il travaille avec sa connaissance : il raisonne sur LE cas qui se présente et il le résout. Après, il l'oublie et passe à un autre ou à autre chose. Ce n'est pas fatigant. Que fait le programmeur une fois qu'il a compris la connaissance de l'expert à mettre dans le programme : il raisonne sur l'ensemble des cas qui peuvent se présenter au programme et il code chacun d'eux ! Quand on sait qu'il peut y en avoir des centaines ou des milliers, parfois beaucoup plus, on comprend qu'écrire un programme réclame beaucoup de temps.

Que fait le système expert de type Pandora, lui ? Il raisonne sur LE cas qui lui est présenté, il le résout et il l'oublie. Exactement comme l'expert. Pourquoi le programmeur se casserait-il les pieds à imaginer puis créer une procédure traitant de tous les cas possibles si le programme peut inventer immédiatement la solution pour chaque cas ?

● 4^{ème} défaut du procédural : *le programmeur fait un boulot de dingue ...et il en est fier*

Non seulement le pauvre programmeur est obligé de comprendre une expertise qui n'est pas la sienne, non seulement il doit imaginer tous les cas possibles, mais encore il doit suivre une procédure complexe s'il veut être sûr que son programme tourne et soit accepté : analyse, ordinogramme, codage, tests/modifications. Plus toutes les étapes nécessaires à l'intégration définitive du nouveau programme dans le système informatique : présentation du programme aux utilisateurs, prises en compte des demandes de modifications des utilisateurs, modifications de l'ordinogramme, re-codages, re-tests/modifications, re-présentations du programme aux utilisateurs, etc. Je dis « etc. » car ces allers-retours entre l'utilisateur non satisfait du programme et le bureau du programmeur peuvent se compter par dizaines. Et je ne parle pas du travail de mises à jour sur les programmes, tâche encore plus complexe qui prend un temps énorme. En effet, le programmeur doit relire des pages de codes dont il ne se souvient plus et, pire, que souvent il n'a pas écrites lui-même. Cette re-programmation est une tâche très stressante pour lui : il s'agit de modifier un outil maintenant fiabilisé et utilisé couramment par des gens qui ne peuvent s'en passer et il sait d'avance qu'il va le leur livrer avec des bugs... Bonjour l'ambiance !

Ceci dit, il est conscient que personne ne peut mettre le nez dans un tel travail, que les utilisateurs n'y comprennent rien et qu'il est de ce fait incontournable... L'évolution des langages informatiques dans le temps est le signe tangible qu'en matière d'informatique, le pouvoir n'est pas entre les mains de ceux qui l'utilisent. *Tout indique qu'il faut plus longtemps aujourd'hui qu'hier pour*

développer un programme et que ça s'aggrave sans cesse. Un langage comme Java, très récent, plébiscité par la communauté des informaticiens au point que la majorité des projets nouveaux sont prévus avec lui, réclame davantage de temps en développement que le langage « C » plus ancien ! Les services utilisateurs d'entreprises se rendent bien compte de cette dérive et font de plus en plus pression pour maîtriser eux-mêmes les développements. Encore faudrait-il qu'ils soient tenus au courant des progrès réalisés pour eux...

Avec la Maïeutique, l'ordinateur fait déjà 50 % du travail, ce qui est un gage de rapidité et de fiabilité. L'expert n'a à réfléchir que sur son savoir-faire, pas sur le futur programme. Les utilisateurs et lui voient le programme s'élaborer sous leurs yeux. Ils peuvent rectifier au fur et à mesure si nécessaire. Il n'y a plus besoin de la présence du programmeur et de toutes les étapes imposées par le procédural. Ca va au moins 10 fois plus vite. La mise à jour des programmes est encore plus facile que le développement, la connaissance étant déjà structurée dans un certain nombre de pages identifiées. Sauf que là, ce n'est plus l'ordinateur qui guide l'expert, c'est l'expert qui guide l'ordinateur : il cherche la page où se trouve la partie de son savoir-faire à modifier, la lit (ce sont des arbres de décision écrits en langage courant, vous allez voir plus loin) et modifie l'endroit désiré. L'ordinateur re-génère l'ensemble du programme, en quelques secondes, et fournit un nouveau programme totalement fiable. Avec la Maïeutique, l'ordinateur est au service de l'utilisateur, pas l'inverse !

● 5^{ème} défaut : *pendant tout le développement, l'expert ne voit pas son programme !*

Quand l'expert (ou un utilisateur) demande la réalisation d'un programme, il ne peut dire en général au programmeur quelle tête ce programme devra avoir à la fin. Ce n'est pas son métier d'imaginer le programme en train de tourner alors qu'il ne sait déjà pas de quoi est faite sa connaissance ! Quant au programmeur, bien qu'il soit expert en informatique, il ne peut pas lui dire non plus à quoi ressemblera le programme à la fin : il n'a aucune idée de l'expertise à recueillir. Son métier c'est informaticien, pas expert-comptable ! Il ne sait pas non plus comment les collègues de l'expert travaillent actuellement avec cette expertise et donc *il ne peut décrire le programme qu'il serait bon de leur proposer pour que ce soit mieux.* C'est un dialogue de sourds. Tout ce petit monde n'aura une idée du programme que ...lorsqu'il sera terminé ! C'est à dire bien tard. Et là, les utilisateurs ne seront pas d'accord, l'expert dira qu'il y a des erreurs ...d'où les allers-retours.

Avec la Maïeutique, *no problem* : le programme est développé dans le service concerné, par les personnes concernées, qui le voient évoluer sous leurs yeux et peuvent le tester en permanence « pour voir ». Ils le rectifient ensemble autant de fois que nécessaire. Quand il est fini, il est fini. Il n'y a plus qu'à le transmettre au service informatique pour intégration avec les autres programmes existants.

● 6^{ème} défaut du procédural : *on ne touche plus au programme une fois écrit*

Il faut tellement de temps pour modifier un programme et sa fiabilité devient alors tellement incertaine qu'on ne le modifie que le plus rarement possible : tous les 6 mois, tous les ans... Un tel délai ne convient en général pas aux services utilisateurs, mais c'est comme ça...

Avec la Maïeutique, tous les programmes peuvent être modifiés *quotidiennement s'il le faut.* Ce sont des bases de données, ce qu'il y a de plus fiable et de plus facile à modifier en informatique. Du coup, les mises à jour peuvent être quotidiennes.

● 7^{ème} défaut du procédural : impossible de modifier un programme en cours d'utilisation

Avec Le procédural, on ne sait pas dans quel état sont les données ni où en était la procédure si on remplace brutalement la version courante du programme en cours d'utilisation par une version ultérieure. L'exécution ne peut donc repartir du bon endroit. Les utilisateurs devront recommencer le travail, en plus sans savoir à partir de quand ! C'est pourquoi la version d'un programme procédural n'est remplacée par une autre que la nuit, lorsque tous les utilisateurs dorment, ou alors pendant une période où leur utilisation est interdite : « Désolé, cette application est en cours de maintenance, veuillez attendre SVP... ».

Au contraire, un système expert de type Pandora peut être remplacé par une nouvelle version en cours d'utilisation, de la même façon qu'une base de données ...puisque c'en est une ! En effet, ce qui évolue constamment dans un système expert, c'est sa base de données de connaissances : la base de règles. Sur Internet, ce type d'opération est fréquent : on l'appelle « mise à jour de bases de données en temps réel ».

● 8^{ème} défaut du procédural : impossible d'aborder certains domaines d'applications

Aujourd'hui, à cause de tous ces défauts, toute un domaine d'applications n'est jamais traité. C'est d'abord celui où règne « l'explosion combinatoire » : le nombre de cas à traiter dans le programme devient tellement énorme que l'on sait d'avance - théorème de Gödel oblige - que l'on n'arrivera jamais à un programme fiable. C'est ensuite celui des applications qui évoluent fréquemment pour ne pas devenir obsolètes, ce que Le procédural ne sait pas faire vu le temps qu'elle met déjà à réussir une mise à jour fiable (6^{ème} défaut).

Ce domaine d'applications est le suivant :

- les Conversationnels car 1) c'est le règne de l'explosion combinatoire et 2) ils reposent sur des connaissances pointues sans cesse affinées et il faut les faire évoluer constamment
- les aides au diagnostic et les aides à la décision, sujettes à l'explosion combinatoire et qui doivent s'améliorer sans cesse en fonction du retour d'expérience
- les logiciels fournissant des conseils sur des prestations évoluant souvent, les logiciels d'aide à la configuration de machines complexes
- les logiciels réclamant essentiellement du raisonnement sur des cas complexes : simulation logique de machines ou de systèmes, test d'idées, diagnostic de panne.

La solution pour traiter la complexité, c'est le raisonnement sur la connaissance comme nous savons tous le faire. C'est le domaine idéal de la Maïeutique. C'est d'ailleurs bien par là qu'elle a commencé, en 1986...