



**HAL**  
open science

## Mieux penser les interfaces informatiques

Michel Beaudouin-Lafon

► **To cite this version:**

| Michel Beaudouin-Lafon. Mieux penser les interfaces informatiques. 2016, pp.61. hal-01435924

**HAL Id: hal-01435924**

**<https://hal.science/hal-01435924>**

Submitted on 8 Feb 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Mieux penser les interfaces informatiques

---

Michel Beaudouin-Lafon, Université Paris-Sud

*Qui n'a jamais été frustré, voire énervé face à un logiciel qui ne fait pas ce que l'on veut ou une borne interactive récalcitrante ? Comme tout outil, l'ordinateur est destiné à la fois à nous simplifier la vie et à augmenter nos capacités. Mais faire simple est parfois compliqué !*

Nos interactions avec le monde numérique passent nécessairement par des interfaces: hier le clavier et la souris, aujourd'hui les écrans tactiles demain les gestes et la voix. Concevoir ces interfaces, c'est le domaine de l'interaction homme-machine (IHM). Ce domaine est presque aussi ancien que l'informatique elle-même car quel que soit le système informatique, il implique un être humain. Pour prendre un exemple extrême, un système embarqué comme celui qui contrôle le gonflement d'un airbag en cas d'accident, a été conçu en fonction de la physiologie humaine, pour éviter de blesser le conducteur ou le passager en se déployant. Et si certains logiciels n'ont pas besoin d'interface pour leur utilisation normale, ils en sont certainement munis pour aider les développeurs à surveiller leur fonctionnement et à les dépanner en cas de problème : aujourd'hui, un garagiste branche son ordinateur sur votre voiture pour la « *debugger* », plutôt que d'utiliser la clé à molette.

## Prévoir les usages non prévus

Pourtant, si les interfaces actuelles paraissent parfois complexes, déroutantes et contre-intuitives c'est parce que bien souvent, on laisse le programmeur s'improviser concepteur d'interface. Or concevoir des interactions simples, naturelles, intuitives, relève d'une science, l'IHM, qui est loin d'être une science exacte. Il faut faire appel aux connaissances de la psychologie sur l'être humain, mais aussi de la physiologie ou de la sociologie pour créer des interfaces qui soient adaptées à nos capacités de perception, d'action et de cognition.

Et il faut aussi prendre en compte la faculté d'adaptation de l'être humain, pour le meilleur comme pour le pire. Confronté à un système inutilement compliqué mais dont nous avons besoin, nous allons nous y adapter, devenir expert, et le trouver presque normal. Au point que si on propose de le remplacer par un système plus simple et plus efficace, on ne voudra pas en changer car cela implique de ré-apprendre des automatismes.

Un concepteur d'interface est donc confronté à une myriade de possibilités et d'usages non anticipés qui vont affecter la qualité de l'interaction. Celle-ci va aussi dépendre d'aléas techniques et contextuels. Telle animation fluide qui permet de comprendre l'interface devient trompeuse lorsqu'elle est saccadée. Une interface efficace à la souris devient fastidieuse sur un écran tactile, etc. Concevoir pour des usages et dans des conditions non prévus, voilà le défi !

La simplicité et l'ergonomie ont bien un coût, mais elles ont aussi une valeur. Ainsi, les sites marchands ont à cœur de simplifier l'acte d'achat - un point tellement important qu'Amazon détient un brevet sur l'achat « *1-click* », pour lequel de nombreuses entreprises lui paient des royalties conséquentes. Un autre exemple est le site de paiement des impôts du Ministère des Finances : c'est presque devenu un plaisir de payer ses impôts par SMS ou *flashcode* ! Par contre l'enregistrement et le suivi d'une réclamation reste un parcours du combattant...

*Article paru dans CNRS Le Journal, 13 Mai 2016*

*<https://lejournal.cnrs.fr/billets/mieux-penser-les-interfaces-informatiques>*

## Rendre les logiciels moins rigides

Le chemin est encore long pour que « faire simplement les choses simples » soit une réalité dans le monde du logiciel interactif. Au lieu de juger les logiciels sur leur nombre de fonctionnalités – dont la majorité des utilisateurs ne connaissent qu’une infime partie – il vaudrait mieux mettre l’accent sur leur *utilisabilité* et sur le respect de règles de base : montrer l’état du système, rendre visibles les dépendances, utiliser des actions simples pour les commandes courantes, etc.

Mais faire simple, ce n'est pas suffisant ! Lorsque l'on utilise régulièrement un logiciel, on veut pouvoir devenir expert, c'est-à-dire être plus efficace et avoir accès à des fonctions plus avancées. Là aussi, les logiciels actuels sont source de frustration : ou bien ils sont limités, ou bien ils demandent un apprentissage long et fastidieux. Au contraire d’un instrument de musique, dont la pratique répétée permet d’augmenter la maîtrise, les logiciels facilitent rarement un apprentissage progressif. Sans compter les changements de version qui nous obligent à ré-apprendre ce que l’on savait déjà faire...

Un outil puissant, c'est un outil qu'on peut détourner, adapter, façonner à sa main. Mais le logiciel est rigide, il ne fait que ce pour quoi il a été programmé. On peut coller une image dans un cahier qui a été fait pour prendre des notes, mais allez essayer de mettre une image dans un logiciel qui ne connaît que le texte ! Il nous faut des logiciels – et des interfaces – plus flexibles, plus malléables, plus modulaires.

Alors que le coût du logiciel est presque exclusivement un coût de conception, et que le coût de manufacture est quasi-nul, comment se fait-il qu’un seul traitement de texte ait acquis un quasi-monopole ? Comment imaginer qu’un seul logiciel puisse être optimisé pour la secrétaire qui tape des mémos, l’écrivain qui écrit un roman, le scientifique qui rédige un article ou le spécialiste qui écrit une documentation technique ? Plutôt qu’un couteau suisse qui permet de faire un peu tout, mieux vaudrait pouvoir choisir une sélection d’outils optimisés pour un usage défini.

## Concevoir des usages plutôt que des fonctionnalités

Alors que pouvons-nous faire, nous, utilisateurs de ces logiciels ? Nous devrions d’abord exiger l’interopérabilité pour ne plus être prisonnier d’un éditeur de logiciel, d’une application ou d’un service en ligne. Interopérabilité et modularité stimuleront naturellement la compétition pour offrir des alternatives et créer de nouvelles interfaces qui exploiteront plus rapidement les résultats de la recherche en IHM. Rappelons que les interfaces actuelles n’ont presque pas changé depuis 30 ans !

Nous devrions aussi exiger que les essais des logiciels évaluent leur utilisabilité, y compris avec des critères objectifs et mesurables. Le *benchmark* d'un processeur n'a aujourd'hui aucun intérêt pour un ordinateur de bureau ou un portable. Mais un benchmark du temps qu'il faut pour renommer 50 fichiers, importer les photos de son smartphone, ou faire un mailing personnalisé avec un attachement, pour ne prendre que quelques exemples, serait sans doute très instructif. Ce que les éditeurs de logiciels doivent comprendre, c’est qu’il leur faut concevoir des interactions, pas seulement des interfaces, il leur faut créer des usages, pas seulement implémenter des listes de fonctionnalités.