

# Magic Ink, de Bret Victor

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 juillet 2007

<https://www.bortzmeyer.org/magic-ink.html>

---

Un passionnant essai consacré à la présentation de données quantitatives par un logiciel et à la meilleure façon dont on peut permettre la navigation dans ces données.

Bret Victor, auteur de ce *"Magic Ink : Information Software and the Graphical Interface"* <<http://worrydream.com/MagicInk/>> cite abondamment l'expert Edward Tufte qui, dans ses livres comme *"Visual Explanations"* <<https://www.bortzmeyer.org/tufte-visual-explanations.html>> explique comment visualiser des données numériques de façon à permettre leur analyse par l'utilisateur. Tufte se limitait surtout aux affichages statiques, comme le papier, et Victor s'attaque aux possibilités supplémentaires qu'offre le logiciel.

Sa thèse principale est qu'on a trop mis l'accent sur l'**interactivité**. Sous couvert de donner le contrôle à l'utilisateur, on lui impose une intense gymnastique (cliquer, dérouler, sélectionner) pour accéder à une information comme « À quelle heure part le prochain train pour Richmond ? ».

À travers cet exemple, celui du BART, le réseau ferroviaire de la baie de San Francisco, Victor nous fait suivre le développement de son logiciel *"BART widget"* <<http://worrydream.com/bartwidget/>>, qui tente de mettre en œuvre ses idées sur l'interface utilisateur. *"BART widget"* tente de **prédire** ce que veut l'utilisateur (par exemple, on planifie rarement ses voyages en métro des semaines à l'avance, donc on peut supposer que l'utilisateur veut un train tout de suite) et, en utilisant les leçons de Tufte, de présenter le plus d'informations possibles, sans que l'utilisateur n'aie à le demander, et en collant aux questions qu'il se pose, pas aux données sous-jacentes. (Victor compare, défavorablement, le site Web officiel du BART à son logiciel.)

Allant plus loin que ce que permettent la plupart des environnements de développement logiciel existants, Victor imagine des systèmes où les logiciels communiquent facilement entre eux, ce qui permettrait à chaque application d'utiliser le **contexte** dans lequel se trouve l'utilisateur pour le servir plus intelligemment. Il cite l'exemple d'un utilisateur qui cherche le mot « chat » dans un moteur de recherche. Si l'utilisateur vient d'éditer un article sur l'Égypte antique, il ne s'intéresse sans doute pas aux mêmes aspects des félins que s'il vient d'éditer un article sur les maladies des mammifères. Notons que Victor parle à peine des risques pour la vie privée qu'entraînent cette communication intensive de

données sensibles (si tous les logiciels sont locaux, ce n'est pas trop grave mais si on communique ces informations à l'extérieur...).

Victor explique ensuite longuement comment on pourrait créer un outil permettant de programmer par exemples : le concepteur pourrait faire des dessins d'un certain nombre de cas et le logiciel inférerait les autres dessins qu'il doit faire, à partir des données. C'est la partie la plus informatique de cet essai, très proche des idées de la programmation fonctionnelle, puisque tout repose sur des simples fonctions, sans effet de bord, qui font correspondre des graphiques aux données numériques.

Tout au long de l'essai, le ton est alerte, les pétillances intellectuelles nombreuses et le programmeur, comme le graphiste, sont certains d'y apprendre beaucoup.

Évidemment, Victor ne va pas faire l'unanimité, d'autant plus que tous les problèmes ne sont pas aussi simples que les horaires de chemin de fer! Par exemple, la discussion sur l'interactivité <http://lambda-the-ultimate.org/node/2167> de l'excellent blog "*Lambda the Ultimate*" <http://lambda-the-ultimate.org/> ne montre pas que de l'approbation enthousiaste.

Dans le secteur des chemins de fer, un connaisseur, Bertrand Petit me fait remarquer que les chemins de fer allemands fournissent des horaires très bien présentés <http://reiseauskunft.bahn.de/bin/query.exe/f> sur leur site Web, même s'ils ne suivent pas les préceptes de Bert Victor, et que le site du BART ne représente pas forcément l'état de l'art. Bertrand cite aussi les suisses, qui ajoutent la possibilité <http://www.sbb.ch/fr/> de fabriquer des fiches horaires papier personnalisées (en PDF) et la possibilité de charger les horaires d'une ligne sur un PDA. Les services allemands et suisses sont tous deux des exploitations du logiciel Hafas [http://www.hacon.de/hafas\\_e/index.shtml](http://www.hacon.de/hafas_e/index.shtml), qui est, selon Bertrand, un bon exemple de l'utilisation intelligente, sur le Web, des principes de conception d'un indicateur des chemins de fer.

Erwan David (auteur d'un plan de voies <http://www.rail.eu.org/plan-voies.txt> en ASCII art) préfère quant à lui l'indicateur Chaix, qui a commencé en 1846 et n'a jamais été surpassé.