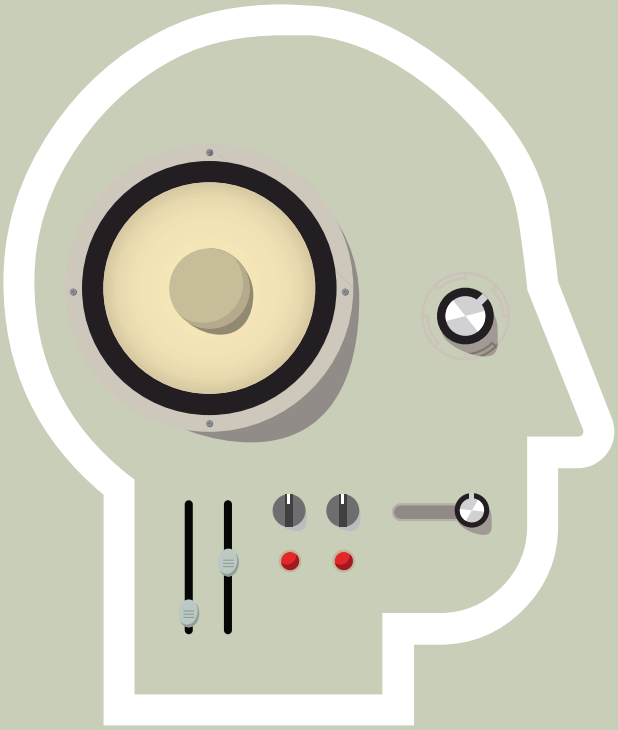
UN ÉVÉNEMENT radio france RADIO FRANCE

Cycle MUSIQUE & CERVEAU

SAMEDI 13 JUIN 2015
MAISON DE LA RADIO



MUSIQUE, MÉMOIRE & APPRENTISSAGE

Grand témoin : ANTOINE HERVÉ

Avec la participation de

Francis Eustache > **La mémoire et le cerveau, des amnésies aux techniques de neuroimagerie**
et Olivier Houdé > **Le cerveau de l'apprentissage - Qu'est-ce qu'un cerveau qui apprend bien ?**
Animé par Hervé Platel > **La mémoire musicale est-elle spéciale ?**

Prochaine conférence

Samedi 12 Septembre 2015
Musique et santé.
Grand témoin : JULIETTE

Avec la participation de **Pierre Lemarquis** > **Soigner par la musique - Histoire et hypothèses**
et de **Daniele Schön** > **La musique au secours du langage – Dyslexie et surdité**
Animé par **Hervé Platel** > **Musicothérapie et cerveau**

Musique et Santé
Mail : info@musique-sante.org
Tél : 04 94 07 76 19
Inscription :
http://www.musique-sante.org/

Coût :
Prise en charge formation continue/OPCA
- Par conférence : 95 €
- Pour le cycle de trois conférences : 240 €

Prise en charge individuelle :
- Par conférence : 80 €
- Pour le cycle de trois conférences : 200 €

Association loi 1901, non assujettie à la TVA N°d'organisme de formation professionnelle : 11 75 317 04 75 N°Siret 421 942 863 000 34

Cerveau & Psycho · ICM Institut du Cerveau et de la Moelle épinière · OCIRP unis par excellence · KLESIA PROTECTION ET INNOVATION SOCIALES · Fondazione Pierfranco e Luisa Mariani neurologia infantile · un lieu universcience · Inserm · SACEM UNIVERSITÉ

HOMO SAPIENS INFORMATICUS chronique de Gilles Dowek

Débattre ou argumenter

Tester un programme ou démontrer qu'il est correct ? La seconde voie, plus satisfaisante, est à privilégier. Pas seulement en informatique…

Les informaticiens ont inventé de nombreuses méthodes pour éviter que leurs programmes comportent trop d'erreurs. Une première consiste, après avoir écrit un programme, à le tester sur quelques centaines ou milliers de cas. Par exemple, un programme qui cherche la sortie d'un labyrinthe étant achevé, on peut le tester sur quelques centaines ou milliers de labyrinthes différents. Si le programme fait, à chaque fois, ce que l'on attend de lui, on acquiert une certaine confiance inductive dans sa correction. Mais comme on ne peut pas tester tous les cas possibles, rien n'empêche qu'un jour apparaisse quand même une erreur.

Une deuxième méthode consiste à démontrer que le programme proposé est correct, c'est-à-dire à construire une preuve mathématique que, dans tous les cas imaginables, il fera ce que l'on attend de lui. Par exemple, après avoir écrit un programme qui cherche la sortie d'un labyrinthe, il s'agit de démontrer que ce programme trouvera toujours une sortie, s'il en existe une. Démontrer la correction d'un programme est plus difficile que le tester, mais, quand on y parvient, on est à peu près sûr qu'aucun bug ne se manifestera lorsque le programme sera en exploitation.

La notion de démonstration rapproche l'informatique des mathématiques, mais la notion de test l'en éloigne. En mathématiques, si l'on veut se convaincre que, par exemple, le résultat de la multiplication $p \times q$ de deux nombres p et q est toujours identique à celui de la multiplication $q \times p$, la seule méthode possible est la démonstration. Bien entendu, rien n'interdit de tester quelques cas tels que 3×7 et 7×3, 12×25 et 25×12, etc. pour se familiariser avec le problème, ou pour formuler une conjecture, mais ces tests ne contribuent jamais à un jugement de vérité.

Contrairement à une démonstration, qui implique une seule personne, le test d'un programme peut être vu comme un dialogue entre deux personnes fictives : l'auteur du programme, qui affirme que son programme est correct, mais sans expliquer pourquoi, et le testeur, qui objecte à cette affirmation une situation où le programme sera éventuellement mis en échec. Si ce n'est pas le cas, l'objection est écartée. Et si toutes les objections du testeur sont écartées, le programme est réputé correct. Une démonstration, en revanche, écarte d'avance non seulement les objections d'un hypothétique testeur, mais toutes les objections possibles.

Il ne suffit pas d'écarter les objections de ses interlocuteurs pour avoir raison.

Le test repose donc sur un étrange renversement de la charge de la preuve : ce n'est plus l'auteur du programme qui doit expliquer pourquoi ce dernier est correct, mais c'est son interlocuteur qui doit trouver une situation dans laquelle l'affirmation du programmeur est mise en échec.

Cependant, ne pas être mis en échec par son interlocuteur ne signifie pas que l'on a raison ; cela peut aussi signifier que l'interlocuteur n'a pas trouvé la bonne objection. C'est pourquoi le fait qu'un programme ait été testé n'exclut jamais qu'une erreur se produise, un jour, dans une situation non explorée par les tests. Et c'est pourquoi la méthode mathématique ne consiste pas à tenir n'importe quelle proposition pour « vraie jusqu'à preuve du contraire » ; de façon générale, la démarche scientifique n'accepte les arguments inductifs que s'il n'est pas possible d'avancer des arguments déductifs.

Cela nous mène à regarder d'un œil suspect la prééminence que notre société accorde aux méthodes dialogiques, et notamment au débat contradictoire, au détriment des méthodes logiques. Comme si nous feignions de croire qu'écarter les objections de ses interlocuteurs suffisait pour avoir raison.

Avant une élection, par exemple, au lieu d'organiser un débat entre les candidats, pourquoi ne les laisserions-nous pas plutôt étayer leurs affirmations par des arguments, leur laissant ainsi la charge de la preuve de ce qu'ils affirment ? ■

Gilles DOWEK est chercheur à l'Inria et membre du conseil scientifique de la Société informatique de France.