



Université Panthéon-Assas



ISCC  
Institut des sciences  
de la communication  
CNRS PARIS SORBONNE

---

## Séminaire annuel du Carism 2016-2017

### En partenariat avec l'ISCC

---

#### Décoder des codes

Cécile Méadel, Francesca Musiani et Guillaume Sire

La phrase de Laurence Lessig — « Le code, c'est la loi » (Lessig, 1999, 2006) — a beau être citée par de nombreux chercheurs en sciences sociales, il est rare de trouver une ligne de code citée dans un article. Cela est d'autant plus regrettable qu'il est fait mention « du code » et de son action plutôt que de parler d'un code en particulier et des actions effectuées par celui-là, ses modalités, ses contraintes intrinsèques. Aurait-on l'idée d'étudier le fonctionnement de la justice sans lire et citer les textes de loi, les arrêts et leurs commentaires ?

Le code est un instrument qui agit et fait agir : il est conçu comme un répertoire d'actions, dont la vocation est de permettre aux machines de traiter l'information et de communiquer entre elles. Il est ce qui permet aux programmeurs de commander la machine ; c'est par exemple le langage qui permet à Google Actualités de repérer des contenus, de les catégoriser, d'établir des relations avec les profils et intérêts propres de chaque internaute, etc. Ainsi le code agit et parle, il permet à la fois et en même temps d'émettre et de recevoir, d'agir et de faire agir. « Un langage de programmation est toujours un compromis entre la puissance d'expression et la possibilité d'exécution » (Dowek, 1998, p. 8). Les langages de programmation supposent une *écriture* (Herrenschmidt, 2007), différente de l'écriture des langues naturelles en cela que « le langage de programmation est un langage formel qui ne permet pas de formulations ambiguës et n'accepte qu'une syntaxe parfaite » (Rieder, 2006, p. 243). Il existerait plus de deux mille langages, et plusieurs *niveaux de langue* : les codes se parlent entre eux, et les développeurs coopèrent et communiquent par la médiation des algorithmes. A chacun de ces niveaux correspond un degré de pouvoir, ou en tout cas une possibilité d'agir et de faire agir, tandis que chaque ligne de code porte à la fois la vision et le projet de celui qui a recours à ce langage et de celui qui en a spécifié la syntaxe et la grammaire.

Ainsi sommes-nous amenés à réfléchir à leur diversité et aux éléments qui conduisent à choisir un code plutôt qu'un autre, à leur complémentarité et à leur concurrence. Sous quelles conditions peut-on citer explicitement les algorithmes ; comment peut-on les montrer pour mieux les analyser, les faire voir pour les faire parler et comprendre. Cela nous permettra de nous interroger sur ce qu'ils font à/de l'information, aux/des médias, à/de l'espace social, à/de la politique, à/de la culture. Quelles sont les variables, les constantes, les balises, les commandes ? Que *signifient-elles* ? On pourra aussi se demander, par l'analyse des pratiques situées des acteurs ce que recouvre l'utilisation, souvent interchangeable, dans les discours des termes « informatique », « code », « programmation », « algorithme »... Ces questionnements et les méthodes d'analyse qu'ils impliquent nous paraissent d'autant plus nécessaires qu'il est de plus en plus question de l'apprentissage « du code » à l'école ou de la circulation de ces notions dans les médias, sans que soit soulevée, ou pas assez à notre avis, la question de savoir de quels codes, au juste, il pourrait s'agir.

Comme l'a écrit Mark Marino : « De plus en plus, le code façonne, transforme et limite nos vies, nos relations, notre art, nos cultures et nos institutions. C'est pourquoi il est temps [...] d'aller plus loin que la seule exécution pour commenter, documenter et interpréter » (Marino, 2006). Plusieurs approches viendront nourrir le séminaire, par exemple les travaux sur les

dimensions sociales et politiques des langages de programmation (dans la ligne par exemple de ceux de Couture, 2012 et Cardon, 2015) et leurs dimensions historique (Rieder, 2012), juridique (Sire, 2014), philosophique (Mélès, 2015), épistémologique (Dowek, 1998, Cardon, 2013), sémiologique (Herrenschmidt, 2007), linguistique (Sire, 2016) et esthétique (Paloque-Bergès, 2006). Le point commun des travaux cités est de ne pas parler « du code » mais « des codes » ou « d'un code » en le(s) nommant, et en n'hésitant pas à citer sa/leur syntaxe, à montrer les algorithmes en analysant et en expliquant ce que cela veut dire à un public de chercheurs en sciences sociales dont les connaissances en informatique sont *a priori* limitées. Cela sera également l'occasion de dialoguer avec des chercheurs en informatique, et de problématiser avec eux les enjeux sociaux liés à l'apprentissage des codes informatiques, ainsi que les enjeux liés au pouvoir et à la responsabilité du programmeur dans l'espace public.

## Bibliographie indicative

- Cardon D. (2013), « Dans l'esprit du PageRank. Une enquête sur l'algorithme de Google », *Réseaux*, 1, n°177, p. 63-95.
- Cardon D. (2015), *A quoi rêvent les algorithmes. Nos vies à l'heure des big data*, Paris, Seuil, coll. Cayley J.
- (2002), « The Code is not the Text (unless it is the text) » *Electronic Book Review*.
- Couture, Stéphane. 2012. « Écriture et performativité du code source ». In *Communication numérique et lien social*, sous la dir. de Serge Proulx et Annabelle Klein, p. 168-184. Namur : Presses de l'Université de Namur.
- Cox G., McLean A., Ward A. (2001), « The Aesthetics of Generative Code », *Generative Art conference. Politecnico di Milano, Italy*.
- Dowek G. (1998), *Le langage mathématique et les langages de programmation*, <http://lsv.fr/~dowek/Vulg/langagelangages.pdf>
- Glazier L. P. (2006), « Code As Language », *Leonardo Electronic Almanac*, vol. 14, n°5.
- Hayles K. (2005), *My Mother Was a Computer. Digital Subjects and Literary Texts*, Chicago, The University of Chicago press, 288 p.
- Herrenschmidt C. (2007) *Les trois écritures. Langue, nombre, code*, Paris, Gallimard, 510 p.
- Lessig L. (2000), « Code is law. On Liberty in Cyberspace », *Harvard Magazine*, traduction française sur *Framablog*.
- Lessig L. (2006), *Code 2.0*, New York, Basic Books, 410 p.
- Marino M. C. (2006), « Critical code studies », *Electronic book review*.
- Mélès B. (2004), « Qu'est-ce qu'un langage de programmation ? », <http://baptiste.meles.free.fr/?Les-langages-de-programmation>
- Mélès B. (2015), « Dieu a-t-il programmé le monde en Java ? Monadologie leibnizienne et programmation orientée objet », [http://baptiste.meles.free.fr/site/B.Meles-Leibniz\\_prog\\_objet.pdf](http://baptiste.meles.free.fr/site/B.Meles-Leibniz_prog_objet.pdf)
- Paloque-Bergès C. (2006), *Poétique des codes sur le réseau informatique*, Oxford (Ohio), Critical Documents, 154 p.
- Rieder B. (2012), « What is in PageRank? A Historical and Conceptual Investigation of a Recursive Status Index », *Computational Culture*, 2
- Sire G. (2015), « Inclusion exclue : le code est un contrat léonin. Enquête sur la valeur technique et juridique du protocole robots.txt », *Réseaux*, n°189, p.187-214.
- Sire G. (2016), « Ce que coder veut dire : y a-t-il un langage de programmation ? », *French Journal for Media Research*, n°6, à paraître.