

CE QUE CODER VEUT DIRE : Y A-T-IL UN LANGAGE DE PROGRAMMATION ?

Plaidoyer pour une approche sociolinguistique du code

Guillaume Sire

Maître de Conférences en Sciences de l'information et de la communication
Centre d'Analyse et de Recherche Interdisciplinaires sur les Médias
Institut Français de Presse / Université Paris 2
guillaume.sire@u-paris2.fr

Résumé

Même si le terme « langage de programmation » procède d'une métaphore, les sciences sociales pourraient avoir intérêt à considérer que le code informatique est effectivement un langage. Une telle posture serait notamment susceptible de les aider à mieux appréhender les rapports de pouvoir propres à l'écosystème numérique. Nous plaidons par conséquent pour une approche sociolinguistique du code, laquelle consisterait à analyser un programme comme on analyserait un texte en prêtant attention à sa syntaxe, les mots employés, la tournure des phrases, les interprétations possibles, tout en nous intéressant aux intérêts économiques et symboliques des acteurs concernés, à leurs projets initiaux et aux effets concrets de ce texte dans l'espace social. Nous terminons en expliquant pourquoi ce cadrage épistémologique pourrait permettre aux SIC de comprendre comment, dans quelle mesure et à quel point le web est *négocié*.

Mots-clés :

Code, programmation, algorithmes, Internet, langage, sociolinguistique

Abstract

Even if the term “programming language” comes from a metaphor, it might be useful for digital humanities to consider that code is indeed a language. Such a point of view would allow them to better tackle what the power relations are within the digital ecosystem. We therefore advocate for a sociolinguistic approach of programming. This would consist of analysing code as a text, by taking into account its syntax, chosen words, sentences and possible interpretations without neglecting economic and symbolic dimensions of the studied phenomenon. We eventually explain how such an epistemological framework could allow media studies to understand how, and to which extent, the web is negotiated.

Keywords

Code, software, algorithms, Internet, language, sociolinguistics

Les langages de programmation ont très peu été considérés par les sciences sociales comme *des langages*, ni dans le sens restreint du terme, désignant les seules langues naturelles, ni dans le sens plus large de systèmes de signes (Ducrot et Todorov, 1972, p. 7). Lorsque un fragment de code informatique est citée dans un article à propos des technologies numériques (ce qui est extrêmement rare), c'est en tant que modalité d'agencement computationnel, une manière de signifier à l'ordinateur comment coordonner les impulsions électriques. Le code est alors réifié, comme si ses lignes ne dépendaient pas de ceux dont elles sont le fait — ou comme si elles en dépendaient trop peu pour que cela puisse être intéressant —, et comme si finalement ce que l'on nomme « langage de programmation » désignait une procédure plutôt qu'un idiome, une façon d'exécuter des ordres plutôt que de communiquer des intentions, et de transmettre du sens plutôt que d'en produire.

Si quelques auteurs, notamment dans le champ esthétique, ont bel et bien considéré que le code constituait effectivement un langage, ce fut pour des approches essentiellement littéraires (Cox et al., 2002 ; Glazier, 2006), certains chercheurs comparant même le code informatique aux belles lettres (Hayles, 2005) tandis que d'autres se livraient à diverses tentatives poétiques (Paloque-Bergès, 2006). Mais l'analogie ne donna lieu à aucune approche visant à comprendre les aspects et les effets sociaux de ce prétendu langage. Nous pensons que c'est regrettable, et qu'il pourrait être heuristique de considérer que le code informatique induit un être-ensemble. Il nous semble en effet que cela conduirait à plusieurs implications théoriques et empiriques utiles pour les sciences sociales, et en particulier pour ce que certains nomment aujourd'hui *les humanités digitales*. Comme l'a écrit Mark Marino, l'un des seuls à notre connaissance à avoir plaidé pour une étude critique de ce que font et disent les langages de programmation : « 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 [ce que coder veut dire]. Considérons que le code est un texte » (Marino, 2006).

Nous plaiderons ici pour une approche sociolinguistique du code, dont la méthode consisterait à étudier les variantes des langages de programmation dans le but d'interroger et de comprendre les variantes sociologies qui les ont produites (Ducrot et Todorov, 1972, p. 84). Nous proposons également le processus inverse, devenu classique en sociolinguistique depuis notamment les travaux de Von Humboldt, lequel voyait dans le langage non pas un effet observable du social, aidant à l'expliquer, mais aussi la possibilité d'une cause explicative en soi. Autrement dit, il s'agit de considérer qu'un langage de programmation n'est pas forcément un objet susceptible de nous renseigner à propos du social mais qu'il peut aussi être la cause des dites dynamiques sociales, et qu'il doit par conséquent être étudié non pas simplement dans le but de décrire les aboutissants des phénomènes mais également, et tout autant, pour en comprendre les tenants. Une telle perspective, à notre avis, manque. Et elle manque d'autant plus qu'elle est nécessaire si l'on veut pouvoir réguler efficacement l'écosystème digital, d'une part, et d'autre part si l'on souhaite comprendre comment et dans quelles conditions il convient d'enseigner le code informatique à l'école puis à l'université tout en apprenant aux élèves/étudiants à avoir le recul critique nécessaire vis-à-vis des technologies numériques, comme on le leur apprend vis-à-vis des autres médias, lorsqu'on les sensibilise par exemple aux influences possibles des sources de financement sur le contenu de la presse.

TROIS ECUEILS

Une première classe d'arguments contre le recours non-métaphorique au terme « langage » dans le cas du code informatique tient au fait que le code s'adresse à des machines et non à des êtres humains. Il ne s'adresse donc ni à la vue ni au toucher et serait par conséquent dépourvu de « mythographie » et de « logographie » (Ducrot, Todorov, p. 249-256). C'est ce qui conduit certains chercheurs à la conclusion selon laquelle « parler de langages de programmation pour un ordinateur reste éminemment métaphorique » (Mélès, 2004). Du point de vue de l'anthropologie de la communication, le risque d'une telle assertion est d'oublier la dimension sociale des ordinateurs, alors même que ces derniers n'agissent pas dans un univers qui serait purement technique mais au contraire ont vocation à relier des individus en leur permettant d'interagir. En considérant que des machines peuvent fonctionner indépendamment de ceux qui les ont conçues et de ceux qui les utilisent pour agir, interagir et communiquer, le chercheur intéressé par une dynamique sociale où l'informatique jouerait un rôle décisif risquerait de ne décrypter que partiellement les tenants et les aboutissants du phénomène qu'il prétend étudier, s'il considérait a priori que le code *n'est déterminant que pour les seules machines*.

Un deuxième type d'argument opposé à l'idée que le code puisse être un langage tient au fait qu'il ne sert pas à signifier mais à faire : les machines exécutent des programmes, elles ne communiquent pas. Selon John Caley, penser le contraire relèverait d'une utopie qui « consisterait à considérer que les symboles qui apparaissent à la surface (ce que vous lisez sur l'écran) sont la même chose (et sont donc réductibles au codage numérique) que les symboles utilisés dans le langage de programmation qui stocke, manipule et distribue le texte que vous lisez » (Caley, 2002). Selon un tel point de vue, les lignes de codes qui remplissent la même fonction sont absolument interchangeables même si leurs syntaxes sont totalement différentes. Autrement dit, le choix d'un langage de programmation, ainsi que la façon dont le programmeur utilise ce langage — i.e. son style — n'ont aucune importance du moment que les lignes de code parviennent à leur fin. Le code a vocation à accompagner le sens mais ne le produit pas.

Le dernier type d'argument susceptible de justifier que le terme « langage de programmation » ne désigne rien de plus qu'une métaphore est l'absence d'ambiguïté possible. En effet, il ne faut pas perdre de vue 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). Autrement dit, à la différence d'un langage naturel ou d'un système de signes classique, le langage de programmation ne peut créer d'ambiguïté ou de quiproquo concernant un couple signifiant/signifié.

Ces trois principaux arguments sont intéressants en cela qu'ils nous invitent à la prudence, dès lors qu'ils soulignent le fait que si nous considérons le code comme un langage nous ne pourrions pas pour autant considérer qu'il s'agit du même type de langage que les langages verbaux dits « naturels ». Cependant ils ne nous paraissent pas suffisants pour nous empêcher de considérer que le code est un langage, quand bien même ce serait un langage de type particulier au même titre par exemple que le langage non-verbal. D'une certaine façon, ils semblent même nous y inviter : le premier en nous rappelant que les machines communiquent entre elles (le langage sert à communiquer, il est social), le second en nous rappelant que cette communication induit des actions (le langage fait et fait faire, il est politique), le troisième en précisant que le code laisse peu de place à l'interprétation tout en

nous rappelant de ne pas oublier le caractère technique de ce langage (le langage propose une vision du monde plus ou moins plastique, il est culturel). Ces trois types d'arguments nous mettent en garde également vis-à-vis du caractère romantique de la comparaison entre le code informatique et le langage humain. Ils nous préviennent des écueils, donc, mais nous ne pensons pas pour autant qu'il faille résister à la tentation de nous engouffrer sur la piste qu'ils balisent sans fermer, et voir, ou au moins essayer de voir, si ce que nous y trouvons peut aider à comprendre les dynamiques sociales, politiques et culturelles à l'ère du numérique.

UN LANGAGE QUAND MEME

Le célèbre sociologue Théodore Nelson, présenté en général comme l'un des pères fondateurs d'Internet, à l'origine de la navigation hypertextuelle, a parlé des ordinateurs comme des « machines littéraires » en considérant que la littérature était un « système d'écrits interconnectés » et que « presque tous les écrits faisaient partie d'une littérature » (Nelson, 1993). Il fonda sa réflexion sur une idée évolutive du texte et s'appuya pour cela sur les théories modernes du champ littéraire (Paloque-Bergès, 2006, p. 6). Ainsi, selon une telle perspective, le code pourrait bel et bien produire du sens, en même temps qu'il induirait certaines actions. Il mériterait par conséquent d'être étudié comme un langage, non pas seulement à visée esthétique, mais aussi à visée sociale, dès lors qu'il serait susceptible d'influencer les rapports entre individus et qu'il serait, comme tout langage, à la fois « regard sur le monde » et « action » (Lochard et Soulages, 2006, p. 238).

Le code informatique peut s'apparenter à ce qu'en linguistique on nomme, justement, « un code », car il « procède de l'association entre des signifiés et des formes conventionnelles, ou signes, qui permettent de les manifester » (Auroux, 2010, p. 91). Les « commandes » et les « balises » du code informatique sont autant de signes renvoyant à des signifiés en faisant explicitement référence à des documents, des variables préalablement renseignées ou à certains types d'action. Il serait regrettable de laisser l'étude de ce que disent et font ces couples signifiants/signifiés du seul côté des sciences appliquées sous prétexte que les signifiés doivent être *renseignés* pour que les signifiants *opèrent*, dès lors que ces couples résultent d'une prise en main par les acteurs des possibilités de communication offertes par le code, et dès lors que cette prise en main vise à dire et faire des choses qui changeront, renforceront ou tout du moins questionneront les dynamiques sociales à l'œuvre.

C'est pour considérer ce que la production et la circulation du sens par la médiation du code informatique font de *moi*, de *toi* et de *nous* que nous nous rejoignons Mark Marino lorsqu'il prétend que les sciences sociales auraient beaucoup à apprendre et à apporter en se penchant sur le code : « Tandis que les chercheurs en informatique théorisent la façon dont le code peut être le plus utile, explique-t-il, les sciences sociales pourraient aider à comprendre sa signification » (Marino, 2006). Comme lui, nous croyons que considérer le code comme un langage à part entière (sans oublier qu'il ne s'agit en aucun cas du même type de langage que les langages dits naturels), permettrait d'en comprendre — ou tout du moins d'en interroger — la *portée* sociale, politique et culturelle. Nous pourrions voir un langage de programmation à la fois comme une cause et un effet de l'être-ensemble qu'il induit et qui l'induit, ainsi qu'une fabrique et un véhicule du sens, de la signification.

L'acte de coder, comme l'acte de parler ou d'écrire, et comme n'importe quelle actualisation d'un langage, « pose un contrat avec l'autre dans un réseau de places et de relations » (De Certeau, 1980, p.38). Etudier le code selon une perspective sociolinguistique reviendrait donc à essayer de formuler les termes et les implications de ce contrat, ou de cette « promesse » au sens de Ricoeur (2004, p. 203-214). C'est ce que la célèbre phrase de Lawrence Lessig — le code, c'est la loi (Lessig, 1999, 2006) — suggère implicitement d'étudier : quelle loi ? écrite comment ? avec quels mots ? quelles tournures de phrase ? quel style ? pour quelles interprétations possibles ? par quels législateurs ? quelle jurisprudence ? quel poids normatif ? comment ? où ? pourquoi ? quelles résistances ? quelle police ? quelles sanctions ?

EXEMPLE : ETUDIER LE WEB

La langue maternelle du web est le langage HTML. La syntaxe de celui-ci est fixée dans une arène extrêmement particulière, le World Wide Web Consortium (W3C), organisme à but non-lucratif fondé par l'inventeur du web : Tim Berners-Lee. Ce dernier a créé le W3C pour que « le web réalise tout son potentiel, en développant les protocoles et directives susceptibles d'assurer sa croissance à long terme » (W3C, 2006 *cit. in.* Halpin, 2008). Il s'agit d'un organe d'innovation destiné au web, mais aussi d'un organe de promotion visant à ce que cette méthode soit préférée aux méthodes concurrentes (par exemple, les applications pour mobile). L'une des principales activités du W3C est de mettre au point le code HTML, donc, un format ouvert permettant d'éditer des pages web qui pourront ensuite être lues par des navigateurs, lesquels se chargeront de traduire le code pour le rendre visible et, surtout, lisible, sur l'écran des internautes. Toutes les organisations, de toutes les tailles, les états, les entreprises, les organisations éducatives et les associations sont invitées à devenir membres du W3C, en versant un montant variant selon ses ressources (jusqu'à 59 000 euros pour une grande entreprise en France), et à proposer des modifications de la grammaire existante, ou des nouveautés, puis à participer aux négociations dont leurs propositions feront systématiquement l'objet. Des groupes de travail sont ainsi organisés dont les productions sont publiques et donnent lieu à des listes de diffusion par courriel, auxquelles n'importe qui peut s'abonner pour avoir accès au contenu de la discussion autour d'une spécification ou d'un changement de standard.

La langue HTML, sa syntaxe, son vocabulaire, son champ des possibles, fait ainsi l'objet de discussions constantes entre des acteurs extrêmement variés et dont le contenu est systématiquement rendu public¹. C'est une particularité qui, ajouté au fait que cette langue soit de très loin la plus utilisée du web, en fait un parfait objet pour illustrer notre propos. Il serait en effet dommage d'étudier le web en sciences sociales sans considérer la langue qui permet au couple homme/machine de communiquer avec d'autres couples sociotechniques. Pour étudier tout cela efficacement, il faudrait faire une analyse sociologique détaillée auprès des différents acteurs du W3C (comme si, intéressé par la fabrication et les usages de la langue française, on étudiait d'abord les arcanes de l'Académie française où sont négociés et décidés les changements d'orthographe et de syntaxe). Puis il faudrait voir comment les développeurs se saisissent des possibilités offertes par le HTML : de quelle façon, avec quel style, qu'est-ce qu'ils disent (comme si on étudiait, après l'Académie française, ce que

¹ Pour une explication plus détaillée du fonctionnement du W3C, voir : <http://letrainde13h37.fr/35/le-w3c-vue-interieure/>

deviennent les possibilités qui y ont été édictées dans la bouche et sous la plume des Français, comment ils s'en servent, à quelles fins). Il faudrait également considérer les autres codes concurrents ou complémentaires du format HTML —Flash, Javascript — pour essayer de comprendre quelles sont les différentes frontières linguistiques et pourquoi les développeurs qui maîtrisent plusieurs codes choisissent celui-là plutôt qu'un autre (pourquoi ceux qui parlent le français et l'occitan parlent le français ici et l'occitan là). Il faudrait voir ensuite comment ce que les développeurs ont fait est traduit par les logiciels de navigation, qui sont des médiateurs dès lors que celui qui les paramètre peut choisir de quelle façon sera interprétée tel ou tel élément du code (c'est à partir d'ici que la comparaison avec une langue naturelle, comme le française, trouve sa limite). Puis il faudrait voir ce qui a effectivement été compris par ceux qui, par la médiation de leur navigateur et du réseau, ont eu accès aux lignes de codes émises par les développeurs. Enfin, il faudrait voir comment les membres du W3C réagissent pour changer ou compléter les standards étant donné ce qu'ils sont devenus au contact de cette foule d'acteurs. Nous aurions ainsi une vue d'ensemble sur les enjeux propres au langage HTML, sans nous être restreint à l'étude du seul code mais sans non plus nous être intéressés à la seule sociologie des acteurs. Nous l'aurions fait en citant des lignes de code explicitement afin de les analyser comme des énoncés qui circulent et qui influencent leur environnement en circulant, pour circuler et parce qu'ils circulent.

CONCLUSION

Dès lors que l'on souhaite étudier les enjeux sociaux, politiques et culturels liés aux technologies numériques, nous croyons qu'il est nécessaire de comprendre ce que sont et font les langages de programmation. Nous n'avons en effet pas d'autre choix que de lire le code si nous souhaitons comprendre les nouveaux rapports de forces et si nous voulons aider, en tant que chercheur en sciences de l'information et de la communication, les législateurs et les régulateurs à prendre les mesures adaptées à l'écosystème digital.

Il s'agit de donner toute son ampleur à l'analyse des langages de programmation en prenant à la lettre le mot langage, et en considérant que le code est un texte, une forme de texte, qui dit *quelque chose* et sert de base à des énoncés performatifs, c'est-à-dire à la fois à des énoncés capables de produire du réel et à des énoncés capables de changer, modifier, diminuer, amplifier ou travestir la réalité.

BIBLIOGRAPHIE

- Auroux S. (2010), « De la langue à la parole », in. Dortier J.-F. (dir.) *Introduction aux sciences du langage*, Auxerres, Sciences humaines, p. 91-97.
- Cayley J. (2002), « The Code is not the Text (unless it is the text) » *Electronic Book Review*.
- Cox G., McLean A., Ward A. (2001), « The Aesthetics of Generative Code », *Generative Art conference. Politecnico di Milano, Italy*.
- De Certeau M. (1980), *L'invention du quotidien, I. arts de faire*, Paris, UGE/10-18, 348 p.
- Ducrot O., Todorov T. (1972), *Dictionnaire encyclopédique des sciences du langage*, Paris, Seuil, coll. « Points », 480 p.
- Glazier L. P. (2006), « Code As Language », *Leonardo Electronic Almanac*, vol. 14, n°5.
- Halpin H. (2008), « La souveraineté numérique. L'aristocratie immatérielle du World Wide Web », *Multitudes*, vol. 4, n°35, p. 201-213.
- Hayles K. (2005), *My Mother Was a Computer. Digital Subjects and Literary Texts*, Chicago, The University of Chicago press, 288 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.
- Lochard G., Soulages J.-C. (2006), « Sémiotique, sémiologie et analyse de la communication médiatique », in. Olivesi S. (dir.), *Sciences de l'Information et de la Communication*, Grenoble, Presses Universitaires de Grenoble, p. 229-242.
- 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>
- Nelson T. (1993), *Literary Machines, 93.1*, Sausalito, Mindful Press.
- Paloque-Bergès C. (2006), *Poétique des codes sur le réseau informatique*, Oxford (Ohio), Critical Documents, 154 p.
- Ricoeur P. (2007), « *Parcours de la reconnaissance* », Paris, Gallimard, coll. « Folio Essais ».
- Rieder B. (2006) *Métatechnologies et délégation. Pour un design orienté-société dans l'ère du Web 2.0*, Thèse en Sciences de l'Information et de la Communication, Université Paris 8 – Vincennes-Saint-Denis, 388 p.