

« Rhétorique de la conception » et rhétorique du code en littérature numérique : pour une intelligence numérique.

Philippe Bootz¹, Odile Farge²

¹ Université Paris 8. Vincennes Saint-Denis, France
(philippe.bootz@univ-paris8.fr)

² Chaire UNESCO ITEN. Fondation MSH/Université Paris 8, France
(ofarge@gmail.com)

Keywords: programme, outil-logiciel, littérature numérique, humanisme numérique.

1 Introduction

L'intelligence numérique fait référence à nos fonctions cognitives en situation d'interagir dans un écosystème numérique. Elle aide à construire un modèle culturel et social à travers l'informatique ou le numérique. Un nouveau paradigme semble se dessiner. Que se passe-t-il derrière l'écran et comment l'articulation de l'humain (l'intelligence) et de la technique (le numérique) s'opère-t-elle ? Sommes-nous en mesure de revoir nos représentations pour construire ensemble des technologies émancipatrices et porteuses d'un devenir humaniste ? Telles sont les problématiques que nous souhaitons aborder en nous limitant au champ particulier de la littérature numérique.

Nous considérons que les moyens numériques influent sur notre fonctionnement cognitif et parler d'intelligence numérique revient à mettre l'humain au centre du dispositif numérique, ce qui participe pleinement de l'humanisme numérique. L'humanisme s'est, en effet, historiquement appuyé sur le développement des sciences et des techniques pour faire advenir son projet. L'humanisme numérique vise donc à mettre l'humain au centre, non pas du projet technique, ce qui est du ressort de l'ergonomie et de la philosophie de l'activité, mais de l'espace des signes qu'il contribue à créer. Pour nous, l'intelligence numérique interroge notre façon d'« être-au-monde » grâce aux outils numériques.

Les auteurs de littérature numérique représentent une population particulièrement sensible et réceptive à la problématique du rapport entre l'outil et le processus de création, qui touche à la question de la forme et du contenu, du programme et du texte écran. Le champ de la litté-

ture numérique investit aussi bien l'espace informatique que celui du littéraire, que ce soit du côté du lecteur ou de celui de l'auteur. Aussi, le statut du programme s'en trouve modifié : d'un espace d'instructions-outils, il a progressivement été conçu comme un espace de signes confronté à celui de la langue naturelle.

Dans cette perspective, après avoir présenté la nature des œuvres de littérature numérique auxquelles nous nous intéressons, nous aborderons la question du programme et des outils-logiciels, en montrant comment ceux-ci amènent à redéfinir un rapport à la production numérique, tant au niveau de la réception qu'à celui de la création, en relevant les enjeux culturels, politiques et industriels.

2 La littérature numérique

Nous ne nous intéressons ici qu'aux productions qui travaillent la langue et ne peuvent exister que par la présence d'un programme informatique. Ce peut être des « œuvres programmées » par leurs auteurs ou d'autres qui relèvent de la *code poetry*¹. Ces dernières prennent la littérature comme référent : le programme lui-même constitue un texte-relevant de catégories littéraires bien stabilisées. L'ensemble de ces œuvres sont donc « plus ou moins » numériques et on pourrait les classer en fonction de l'importance qu'elles accordent à la programmation. On pourrait de même les classer en fonction de leur emploi des fonctionnalités du dispositif numérique qu'elles utilisent.

3 La place du programme dans les œuvres

Le cœur de notre problématique est situé dans les œuvres programmées qui produisent des médias lisibles sur écran, médias que nous considérons en général comme seuls constituants de l'œuvre, comme seuls intéressants une lecture esthétique ou littéraire, partant du principe que le lecteur n'a pas à être un spécialiste des langages informatiques comme le suppose la *code poetry* dans laquelle les programmes ne sont pas conçus pour être opérationnels ; autrement dit, ils échappent à la fonction outil de la programmation qui est tout de même sa raison d'être.

¹ On en trouvera un certain nombre sur la page du *source code poetry contest* <http://www.sourcecodepoetry.com> (consulté le 24/02/2016).

Christopher Strachey programme en 1952 *Love Letter*, le premier générateur combinatoire informatique de textes [1]. Il considère le programme comme une extension cognitive de l'auteur, une intelligence déléguée. Ainsi donc, le programme n'a pas à présenter en lui-même de qualités autres qu'informatiques. Il n'est pas accessible au lecteur, il fait partie de la sphère privée de l'auteur qui, lui, ne l'envisage pas comme un objet porteur de significations non-informatiques. Il s'agit d'une vision purement fonctionnelle du programme. Le développement des langages de programmation évolués va modifier cette conception chez les programmeurs. À la suite de Donald Knuth [2], les programmeurs sont de plus en plus sensibles à une esthétique du code. Le programme apparaît comme un outil symbolique opérationnel et un texte écrit dans une langue particulière, celle du langage utilisé. Il possède alors, comme tout texte, des valeurs esthétiques. La *code poetry*, qui se développe fortement depuis 2013, s'inscrit dans cette lecture du programme. Ce n'est pourtant pas cette vision qui nous intéresse ici.

L'humanisme numérique en littérature numérique nous amènerait à sortir le programme d'une façon ou d'une autre du domaine privé de l'auteur pour le transformer en objet à lire. Il ne s'agit pas d'inverser la conception fonctionnelle, de sortir le programme de la sphère privée de l'auteur pour la faire passer dans le domaine privé du lecteur comme le font les œuvres fondées sur l'écrit-lecture [3]. Dans celles-ci, le lecteur, par les manipulations du programme, acquiert un statut d'auteur. Il s'agit plutôt de mettre l'humain au centre de la lecture non-informatique du programme pour tous les acteurs qui y sont confrontés : auteurs et lecteurs compris.

Depuis la fin des années 1980 et l'expansion de l'*open-source*, l'idée que le programme constitue une part lisible de la production par le lecteur au même titre que les médias fait son chemin et les auteurs sont de plus en plus nombreux à ouvrir leur code en libre accès. Néanmoins, cette solution ne donne pas pour autant à lire les propriétés esthétiques non informatiques du programme. Le texte imprimé n'est lisible que parce que nous possédons une machine de lecture indépendante du texte : le livre. De même, le programme ne sera lisible de façon esthétique que si nous possédons une machine de lecture qui en soit indépendante : la lecture du code doit être instrumentée. Nous ne possédons pas aujourd'hui une telle machine de lecture car personne n'en a encore pointé la nécessité.

La question de la forme programmée, et donc de la valeur esthétique non informatique des programmes, fut au cœur des préoccupations du groupe Transitoire Observable². Philippe Bootz [4], l'un des fondateurs de ce groupe, s'attache depuis 2008 à théoriser une méthode de conception et d'analyse de programmes fondée sur des blocs signifiants d'un point de vue esthétique, les « tresses » [5] et non plus sur une logique fonctionnelle. Cette dernière est encapsulée dans la « tresse » qui constitue l'unité minimale signifiante au niveau de l'écriture programmée. Ce modèle est intégré à une modélisation plus large du dispositif littéraire numérique dénommé « modèle procédural » [6]. Ce travail a permis de mettre en évidence divers types de représentations sémiotiques liées au programme [7, 8]. Certaines ne sont présentes que dans le code source et d'autres constituent des figures de rhétorique qui mettent en œuvre deux espaces signifiants : celui du programme et celui de l'écran. Le programme de Marschall [9] en est un bon exemple : son listing est le calligramme d'une locomotive et il écrit cho cho à l'écran. Il existe un rapport métonymique entre la locomotive et cho cho, mais également un rapport métaphorique entre le fonctionnement d'une locomotive et l'exécution du programme : il faut fournir de l'énergie à l'ordinateur pour qu'il écrive et à la locomotive pour qu'elle émette un son. Le processus d'exécution, alors, n'est pas donné à voir comme la simple actualisation d'un résultat potentiellement présent dans le programme, mais comme le moteur expressif qui met en marche les processus intéressants à l'humain. Le programme source n'y apparaît plus comme une délégation de l'intelligence créatrice mais comme une machine symbolique pérenne et inerte en attente de l'activation humaine. Ces méthodes d'analyse ont permis de mettre en évidence des dimensions littéraires qui demeurent virtuelles dans le programme tout comme dans le résultat à l'exécution. Elles nécessitent, pour être atteintes, de déconstruire le programme et d'en jouer séparément les tresses [10], autrement dit de construire une machine qui ne demande pas au lecteur de lire le programme. Cette première approche de ce qui pourrait être un précurseur de « machines à lire » les programmes, participe de la mise en œuvre en littérature numérique d'une nouvelle posture de réception, différente de la lecture écran ou de la simple lecture du code, dénommée une méta-lecture dans le modèle procédural.

² <http://transitoireobs.free.fr/to/> (consulté le 24/02/2016)..

4 La « rhétorique de la conception »

La structure des discours adressée au créateur d'une œuvre par l'interface de l'outil-logiciel annonce ce qu'Odile Farge nomme la « rhétorique de la conception » [11] qui profile des enjeux culturels, politiques et industriels. Nous nommons « outil-logiciels », les langages de programmation et les logiciels de création que les auteurs utilisent. Nous décrivons la « rhétorique de la conception » comme une approche de l'outil-logiciel qui sonde à la fois comment cet outil anticipe sur les pratiques des auteurs d'œuvres numériques, et comment ces auteurs s'en emparent en mobilisant des imaginaires en amont et en aval de l'acte de création. L'auteur se retrouve au centre d'une double approche, à la fois utilisateur et auteur d'un dispositif numérique.

Nous supposons que les outils-logiciels portent en eux les codes de création et qu'ils transforment nos modes de raisonnement. Alors que le *Design Pattern* répond aux problématiques de conception de l'outil-logiciel par un ensemble de bonnes pratiques informatiques, nous ne pouvons résumer ou décrire l'outil-logiciel par ces ensembles de fonctions. Prenons l'exemple du logiciel Flash : il offre par ses fonctionnalités un accompagnement dans la tâche d'écriture de l'auteur mais il transforme également sa manière d'écrire, que ce soit dans le style ou dans la relation entre les différents acteurs (sons, vidéos, ...). L'homme, en agissant par le truchement d'un outil, qu'il soit ou non numérique, transforme son environnement et se transforme [12]. Nous dirions alors que l'outil-logiciel a le pouvoir de modifier le sujet que nous sommes. L'*Homo faber*, *Homo sapiens* [13] capable de modifier le monde qui l'entoure, fabriquerait puis manierait des outils qui contiennent déjà une certaine vision du monde, un point de vue inséré dans l'outil et qui entraînerait la modification même de son être.

Il se manifeste dans les outils-logiciels une intention d'adresser un message, dont l'ampleur et l'influence restent souvent inconnues du récepteur. Ce message, structuré selon des stratégies rhétoriques, s'appuie sur des croyances et des sous-entendus. Ils donnent les conditions qui doivent être satisfaites pour qu'un acte soit effectivement accompli. Les « présupposés » de l'outil-logiciel imposent une contrainte implicite à l'utilisateur qui se doit de les accepter, sous peine de ne pouvoir poursuivre le dialogue avec lui. Censés être connus de

l'interlocuteur, ils se présentent comme un « allant-de-soi », désignant ce qui est logiquement impliqué par le maniement de l'outil-logiciel.

La formalisation des discours qui en facilite son intégration, s'effectue « sous l'horizon d'un "impensé" des techniques qui le soutiennent » [14]. Le terme d'« impensé informatique » exprime l'idée qu'un discours efficace contribue à construire une réalité et formate la société où l'homme se trouve fasciné par les outils qu'il emploie. La « rhétorique de la conception » a pour ambition de dévoiler ces enjeux de pouvoir derrière la surface des interfaces, qui préforment les formes et figures que l'utilisateur sera en mesure de créer, favorisant ainsi la diffusion d'esthétiques particulières.

Aussi, l'une des difficultés de la création numérique, littéraire ou autre, repose dans la nécessité d'échapper aux formatages consensuels et la capacité à appréhender un environnement numérique, en apprenant des manières de faire, tout en développant une pensée critique. Autrement dit, l'intentionnalité de l'auteur et sa responsabilité sont engagées : la « rhétorique de la conception » induit un certain partage d'autorité entre l'œuvre et son auteur.

A cet égard, l'utilisation des outils-logiciels dits « libres » et la question du code qui la sous-tend, nous amènent à réfléchir au nouveau paradigme du monde numérique, étroitement lié à la notion d'intelligence collective : la création culturelle autour d'outils ou le code est ouvert et fait partie de l'œuvre, voire constitue l'œuvre. Pour le lecteur, comme l'auteur, il s'agira d'échapper aux stratégies préconfigurées par une industrie, de gagner en autonomie. Accéder au programme revêt pour les auteurs un rôle éminemment politique, l'ultime liberté du citoyen de s'exprimer, de transférer sa représentation du monde, d'apprendre à être indépendant de ceux qui fabriquent les objets, et déployer les valeurs humanistes de partage et de biens communs qui font la force du logiciel libre.

5 Conclusion

Nous assistons à un changement de paradigmes bien plus qu'à un changement de technologies. Les choix de l'auteur et du lecteur conditionnent nos représentations. La dimension sémiotique des outils-logiciels et des programmes ne doit pas être sous-estimée car il s'agit de l'évolution même de nos capacités à aborder ces nouveaux paradigmes.

References

1. Strachey, Christopher. « The 'thinking' machine » *Encounter*, 25-31. [en ligne] <http://www.unz.org/Pub/Encounter-1954oct-00025> (consulté le 10/02/2014). (13, October 1954).
2. Knuth, Donald. *The Art of Computer Programming*, vol.1-4, Addison-Wesley, Reading, Massachusetts (1997).
3. Barbosa, Pedro. *A ciberliteratura Criação Liter'aria e Computador*. Lisboa: Edições Cosmos (1996).
4. Bootz, Philippe. « The Problem of Form: Transitoire Observable, a Laboratory for Emergent Programmed Art ». In *The Aesthetics of Net Literature; Writing, Reading and Playing in Programmable Media*, Transcript Verlag., pp. 89-103. Bielefeld: Gendolla P. & Schäfer J. (2007).
5. Bootz, Philippe. « Using temporal semiotics in a-media digital poetry ». In *Computer Art Congress [CAC.2]; emerging forms of computer art: making the digital sense*, édité par Khaldun Zreik et Garcia Eve Reyes, pp. 5-18. Paris : Europia Productions (2008).
6. Bootz, Philippe. « Haut lieu de l'œuvre au lieu de l'œuvre ». In *E-formes3 : les frontières de l'œuvre numérique*, édité par Alexandra Saemmer et Sophie Lavaud-Forest, pp. 107-115. Saint-Étienne : Publications de l'Université de Saint-Étienne (2015).
7. Bootz, Philippe. « Signs and Apparatus in Digital Poetry: the Example of Jean-Marie Du-tey's Le mange-texte ». *LLC 27*, n°3, pp. 73-92 (2012).
8. Bootz, Philippe et Alexandra Saemmer. « Un éphémère pérenne : dialogue ». In *E-formes3 : les frontières de l'œuvre numérique*, édité par Alexandra Saemmer et Sophie Lavaud-Forest, pp. 279-284. Saint-Étienne: Publications de l'Université de Saint-Étienne (2015).
9. Marshall, Eric. The 3rd International Obfuscated C Code Contest (IOCCC), (1986). <http://www.ioccc.org/years.html#1986> (consulté le 16/11/2015).
10. Bootz, Philippe et Marcel Frémiot. *Passage, poème numérique, éléments d'esthétique et d'analyse*. Les Cahiers du MIM 3. Marseille: MIM (2009). <http://www.labo-mim.org/site/index.php?cahier3bdc> (consulté le 10/02/2010).
11. Farge, Odile. « Rhétorique de la conception » : pour une prise de conscience des stratégies de l'outil de création. Proposition d'une typologie de postures d'auteurs. Thèse de doctorat en Sciences de l'information et de la communication. Université Paris 8 (2014).
12. Farge, Odile. « La "rhétorique de la conception". Pour une conscientisation du rôle de l'outil dans la formation d'une culture numérique ». Dans *Interfaces Numériques*. Volume 4 – No.3, pp. 525-540. Paris : Éditions Lavoisier (2015).
13. Sigaut, François. *Comment Homo devint faber*. Paris : CNRS Éditions (2012).
14. Robert, Pascal. *L'impensé informatique*. Paris : Éditions des archives contemporaines, p. 109 (2012).