

Le Markdown, une praxis énonciative du numérique

< Patrick Mpondo-Dicka ¹ >

1. LERASS (Laboratoire d'Études et de Recherches Appliquées en Sciences Sociales, EA827)
5 Allées A. Machado, Université de Toulouse, Jean-Jaurès, Toulouse, France
patrick.mpondo-dicka@univ-tlse2.fr

DOI:10.3199/RIN.1.1-n © AFDI 2012 IN_DOI

< RESUME >

Le Markdown est un lieu privilégié d'observation de ce que la théorie opérationnelle de l'écriture numérique (Crozat *et al.*, 2011) appelle « *la tension de la manifestation* », c'est-à-dire la tension entre le niveau techno-applicatif des formats, des langages de programmation et des programmes, et le niveau sémio-rhétorique des interfaces. Nous choisirons l'exemple de l'avènement du Markdown, de son évolution, pour montrer, en acte, la praxis énonciative qui préside à ce lieu du dialogue humain-machine.

< ABSTRACT >

Markdown is a privileged place to observe what the operational theory of digital writing (Crozat *et al.*, 2011) calls "the tension of manifestation", *i.e.* the tension between the techno-applicative level of formats, programming languages and programs, and the semio-rhetorical level of interfaces. We will choose the example of the advent of Markdown, its evolution, to show, in act, the enunciative praxis that presides over this place of human-machine dialogue.

<MOTS-CLES>

MARKDOWN, ECRITURE NUMERIQUE, IDIOLECTIQUE NUMERIQUE, DIALOGUE HUMAIN-MACHINE.

<KEYWORDS>

MARKDOWN, DIGITAL WRITING, DIGITAL IDIOLECT, HUMAN-MACHINE DIALOGUE.

1. Introduction

En nous appuyant principalement sur la théorie opérationnelle de l'écriture numérique (Crozat *et al.*, 2011)⁴⁰ développée au COSTECH (Connaissance, Organisation et Systèmes TECHniques) de l'Université de Technologie de Compiègne, nous voudrions mettre en lumière les tensions et les interactions entre les niveaux techno-applicatif et sémio-rhétorique de l'écriture numérique, telle qu'elle est définie par Crozat *et al.*, tensions qui intéressent particulièrement une sémiotique du numérique à laquelle nous essayons de contribuer. Nous appuierons notre réflexion sur un exemple de langage de balisage léger, le Markdown, en analysant les choix présidant à sa conception et son évolution, considérant un tel langage, libre et accessible, comme un lieu privilégié de l'observation du dialogue humain-machine.

2. Une théorie de l'écriture numérique

L'équipe de recherche du COSTECH développe depuis presque une dizaine d'années une « *théorie opérationnelle de l'écriture numérique* » qui propose « *un modèle à finalité heuristique visant à fournir une représentation des propriétés du numérique, et plus particulièrement de l'écriture numérique* » (Cailleau *et al.*, 2012, p. 38) articulée en trois niveaux, « *selon le niveau de concrétisation où on le considère, depuis son idéalité numérique jusqu'à sa réalité sémiotique ou matérielle* » (Croizat *et al.*, 2011) : le niveau théorético-idéal (niveau 1), le niveau techno-applicatif (niveau 2) et le niveau sémio-rhétorique (niveau 3). Nous ne traiterons pas ici du niveau 1, étant entendu qu'il « *repose sur la discrétisation et la manipulation d'unités formelles privées de sens (les 0 et les 1 ou n'importe quelles autres unités logiques formelles constituant un alphabet de manipulation)* » (Petit, Bouchardon, 2017, p. 137) ; proprement combinatoire et calculatoire, les opérations de ce niveau, pour aussi fondamentales qu'elles soient pour caractériser le numérique, n'ont qu'un intérêt relatif pour notre propos. En revanche, nous voudrions situer notre réflexion aux deux niveaux

⁴⁰ Désormais TOEN.

suivants, en considérant leur mise en tension comme sémiotiquement signifiante :

« Ces trois niveaux sont en tension entre eux d'une part, et possèdent par eux-mêmes leur propre tension, d'autre part. Le niveau théorique connaît la tension entre le format formel idéal et le substrat physique sous-jacent, c'est la tension de l'implémentation. Le niveau techno-applicatif connaît la tension entre le format de codage et la forme d'interprétation, c'est la tension de la manifestation. Le niveau sémio-rhétorique connaît la tension entre les fonctions du système et la pratique d'usage, c'est la tension de l'interaction. » (Crozat et al., *op. cit.*)

Les sémioticien·nes se sont essentiellement situé·es au niveau sémio-rhétorique des interfaces envisagées comme lieu d'une rhétorique de la réception (Saemmer, 2015) ou d'une expérience sociale et sensible (Pignier et Drouillat, 2015), ou à l'aune de sa productivité écranique (Jeanneret, Souchier, 2005), tirant l'approche vers une sémio-pragmatique des interfaces, voire une socio-sémiotique des interactions techno-médiatisées. Dans les termes de la TOEN, la préoccupation sémiotique s'est essentiellement focalisée sur la « *tension de l'interaction* ».

Nous proposons d'orienter le regard sur la « *tension de la manifestation* » (Crozat et al., 2011), car il nous semble qu'elle expose plus clairement les enjeux du dialogue humain-machine, entendu comme la négociation de l'humain avec la différence radicale de la machine par l'entremise du seul lieu qui nous soit commun : le code, c'est-à-dire le texte hybride adressé autant à la machine qu'à l'humain, ce que Petit et Bouchardon considèrent comme caractéristique de l'écriture numérique : « *C'est la définition même de l'écriture numérique que d'être tendue entre une écriture visible et une écriture invisible, entre une écriture et un programme.* » (Petit, Bouchardon, 2017, p. 138)

En effet, nous considérons avec Samuel Goyet, qu'

*« un ordinateur est radicalement différent d'un humain, tant en degrés qu'en nature. Cela tient à la nature de son écriture et au volume des opérations réalisées. Il constitue donc, pour nous, une altérité. **En revanche, il partage avec nous un espace : le texte.** En effet, depuis le paradigme logique de Von Neumann, programmer, c'est écrire une suite d'instructions à une machine, ce*

code pouvant être lu tant par l'humain que par un ordinateur. C'est donc le code qui constitue un lieu de rencontre, en tant que texte hybride. » (Goyet, 2017, pp. 556-557, nous soulignons)

Ainsi pourrions-nous toucher du doigt ce qui fait la spécificité de l'écriture numérique, son caractère hybride, entre écriture de la machine et écriture humaine, qui est une tension, selon Petit et Bouchardon :

« S'il y a tension, c'est d'abord parce que l'écriture numérique réunit deux mondes jusqu'alors disjoints : le monde de l'écriture et le monde de la machine. S'il y a tension, c'est ensuite parce que les tensions propres à l'écriture (celles, par exemple, entre écriture et parole, écriture et langue, écriture et lecture) sont reposées à nouveau frais par la tension spécifique à l'écriture numérique (soit la tension écriture/programme). » (Petit, Bouchardon, 2017, p. 138)

Nous porterons nos observations sur le Markdown, un langage de balisage léger, accompagné d'un logiciel libre du même nom. L'accès libre au code nous semble une nécessité pour comprendre les enjeux du numérique (Mpondo-Dicka, 2017, p. 111). Sur ce point, nous faisons nôtre la remarque de Richard Stallman :

« Tout programme incorpore des connaissances. Si le programme est propriétaire, il cache, il refuse ces connaissances aux étudiants, et cela va à l'encontre de l'esprit de l'éducation. [...] Le logiciel libre respecte l'esprit de l'éducation en mettant les connaissances à disposition de tous. » (Stallman, 2017, p. 106)

3. Le Markdown, brève histoire d'un langage de balisage léger

3.1. Un langage de balisage léger, pour quoi faire ?

Le Markdown⁴¹ est un langage de balisage léger développé par John Gruber (aidé d'Aaron Schwartz) en 2004. Les langages de balisage léger⁴² permettent d'écrire dans un fichier en texte brut, en ajoutant au texte des balises simplifiées (en général des signes typographiques de ponctuation entourant ou précédant la suite de caractères à modifier⁴³) permettant d'intégrer la mise en forme du texte de manière lisible par un humain, tout au long de l'écriture, avant qu'elle soit convertie dans un autre format. Selon son auteur :

« Markdown est un outil de conversion de texte en HTML pour les rédacteurs web. Markdown vous permet d'écrire en utilisant un format de texte brut facile à lire et à écrire, puis de le convertir en XHTML (ou HTML) structurellement valide. » (Gruber, 2004b, notre traduction)⁴⁴

Chaque LBL a d'abord été pensé pour fluidifier un processus d'écriture récurrent, en évitant de coder à la main les différentes instructions de mise en forme et de structuration : par exemple, le Setext⁴⁵ a été développé par Ian Feldman en 1992 pour faciliter l'écriture de la *newsletter* TidBITS⁴⁶, un périodique technique

⁴¹ Il existe plusieurs graphies du terme ; l'inventeur du langage utilise majoritairement la majuscule, sans distinction claire entre le logiciel et le langage, mais de nombreux développeurs utilisent l'italique et la minuscule pour désigner le langage. Nous signalerons les noms de logiciels en gras. Dans le cas du Markdown, nous différencierons donc le logiciel (**Markdown**) du langage (Markdown), quitte à modifier les citations des auteurs, ce que nous signalerons.

⁴² Désormais LBL.

⁴³ Par exemple, les signes *, _, #, sont utilisés pour la mise en page en Markdown, et *reStructuredText*, en *Textile*, etc.

⁴⁴ Toutes les traductions de l'anglais sont de notre fait, sauf exception explicitement indiquée. Nous l'indiquerons désormais par la mention Nt après le nom de l'auteur.

⁴⁵ Voir l'URL : <https://en.m.wikipedia.org/wiki/Setext> (consulté le 7 juillet 2019).

⁴⁶ Voir l'URL : <https://tidbits.com/> (consulté le 7 juillet 2019).

hebdomadaire distribué alors par mail, le reStructuredText⁴⁷ par David Goodger en 2002 pour faciliter le processus d'écriture de documentation des programmes codés en Python, Textile⁴⁸ par Dean Allen en 2002 pour faciliter la publication avec **TextPattern**⁴⁹, le CMS qu'il développait de manière concomitante après avoir utilisé ses fonctionnalités pour publier son propre site, *Textism*⁵⁰ ; enfin, le Markdown pour faciliter la publication de billets de blogs dans **MovableType**, le moteur de blog utilisé par John Gruber pour sa revue en ligne *DaringFireball*⁵¹. Gruber a principalement en tête le modèle du courrier électronique lorsqu'il conçoit le langage Markdown :

« Bien que la syntaxe de Markdown ait été influencée par plusieurs filtres de texte vers HTML existants - dont Setext, atx, Textile, reStructuredText, Grutatext et EtText - la plus grande source d'inspiration pour la syntaxe de Markdown est le format du texte brut des e-mails. À cette fin, la syntaxe de Markdown est entièrement composée de caractères de ponctuation, lesquels ont été soigneusement choisis de manière à ressembler à ce qu'ils signifient. » (Gruber, 2004d, *op. cit.*, Nt)⁵².

⁴⁷ Voir l'introduction de David Goodger, URL : <http://docutils.sourceforge.net/docs/ref/rst/introduction.html> (consulté le 7 juillet 2019).

⁴⁸ Voir l'URL : <https://textile-lang.com/> (consulté le 7 juillet 2019).

⁴⁹ Voir l'URL : <https://textpattern.com/> (consulté le 7 juillet 2019).

⁵⁰ Il ne reste aujourd'hui du site qu'une page de démonstration de... Textile, voir l'URL : <https://www.booked.net/textism.html> (consulté le 7 juillet 2019).

⁵¹ Voir la présentation du Markdown par John Gruber sur *DaringFireball*, URL : <https://daringfireball.net/projects/Markdown/> (consulté le 13 avril 2019).

⁵² On demandera au lecteur ou à la lectrice ne connaissant pas la syntaxe du Markdown, de se reporter à son exposition par son auteur sur la page web <https://daringfireball.net/projects/Markdown/syntax#overview> (consulté le 13 avril 2019) ; ou à sa traduction intégrale en français réalisée par Michel Fortin : <https://michelf.ca/projets/php-Markdown/syntaxe/> (consulté le 13 avril 2019), qui a porté le Markdown en PHP. Il existe aussi un *Guide du Markdown*, en anglais, qui décrit la syntaxe de base et plusieurs variantes entre lesquelles il permet la comparaison, enfin, plusieurs outils d'écriture en ligne permettent de rapidement prendre en main le langage, comme le *dingus* (logiciel d'entraînement en ligne) créé par Gruber lui-même, <https://daringfireball.net/projects/Markdown/dingus> (consulté le 13 avril 2019).

Il explique avoir été partisan de l'écriture en HTML brut, considérant que l'HTML était facile à apprendre⁵³. Mais après un an d'écriture de *DaringFireball*, il s'est juste rendu compte non seulement de l'usure provoquée par la répétition de tâches de mise en forme (ce qu'il compensa par l'usage d'un moteur de blog), mais également que le HTML pouvait être pénible à écrire et à lire :

« *Même si vous insistez sur le fait qu'il est facile de composer du texte en HTML brut, n'est-ce pas une corvée ? N'est-il pas fastidieux d'écrire "AT&T" au lieu de "AT&T" ? (Sans parler de la nécessité d'encoder les esperluettes dans les URLs ?)* » (Gruber, 2004a, Nt).

Le Markdown vient pallier cette corvée. Mais pour Gruber, il ne s'agit pas de se substituer totalement au HTML, mais de pouvoir l'écrire quand le besoin s'en fait sentir. Ainsi, le Markdown permet d'écrire en HTML : le script de conversion écrit par John Gruber sait reconnaître les balises du HTML sans les traiter, ce qui en fait un langage différent d'autres langages de balisage léger : « HTML est un format de publication ; Markdown est un format d'écriture » (Gruber, 2004a, Nt).

3.2. Un langage pour une pratique

La raison d'être du *Markdown* permet de comprendre ses limitations. Gruber ne fait que construire l'outil qui lui semble adéquat pour les tâches qui le préoccupent, écrire son blog portant sur l'actualité sociale et technique du numérique. Ainsi, par exemple, doit-il parler de et montrer régulièrement du code informatique, ce qui le préoccupera au moment de créer le *Markdown* : « *En plus de faciliter l'inclusion de balises HTML en ligne dans un document, Markdown facilite également l'inclusion de balises HTML d'exemple dans les intervalles et blocs de code.* » (GRUBER, 2004a, Nt). Le *Markdown* correspond donc à un certain nombre de besoins d'écriture que son développeur a implémenté dans un script :

2019) ou le *Markdown Tutorial* : <https://www.markdowntutorial.com/> (consulté le 13 avril 2019).

⁵³ Rappelons que Gruber s'adresse essentiellement à des développeurs, pour qui le HTML n'est que le langage par lequel il faut passer pour afficher sa documentation en ligne, un simple outil relativement trivial.

écrire, relire et mettre en forme sans effort des articles publiés en ligne au format HTML ; c'est pourquoi Gruber néglige certains aspects de l'écriture qui ne lui sont pas utiles (notes, table des matières...). Plus encore, il n'a que faire de la standardisation de son langage : le HTML est le standard, le Markdown n'est qu'un outil pour en faciliter l'écriture :

« L'idée de Markdown est de faciliter la lecture, l'écriture et la révision de la prose. HTML est un format de publication ; Markdown est un format d'écriture. Ainsi, la syntaxe de formatage de Markdown ne traite que les problèmes qui peuvent être transmis en texte brut. » (Gruber, 2004a, Nt)

Nous pouvons tirer une première conclusion de ces éléments : le *Markdown*, comme la plupart des autres langages de balisage léger, n'est que le reflet de la pratique de son auteur, une mise en forme machinique de ses actions, ou, plus généralement, de son comportement. C'est la praxis énonciative de Gruber, ou du moins du Gruber rédacteur de *DaringFireball* qui est intégrée au langage, et convertie pour la machine via le script *Perl*. Cette individuation du dialogue humain-machine n'est pas spécifique aux LBL, si l'on se reporte à l'histoire de l'informatique. Plus les besoins de calcul brut ont été couverts, plus les capacités physiques de la machine ont augmenté, et plus la programmation s'est localisée, orientée vers *l'adaptation à l'humain* programmeur. Les années 1990 marquent ce tournant des langages de programmation créés pour servir les besoins locaux de leurs utilisateurs : Python comme interpréteur de commande pour le système d'exploitation hollandais Amoeba, Ruby de la frustration de son auteur, Yukihiko « Matz » Matsumoto, à l'usage d'autres langages de programmation⁵⁴, Javascript pour renforcer l'adoption de Netscape, etc. On constate que la motivation est souvent liée à une frustration dans l'usage d'un langage précédent, et qu'elle procède de l'adaptation non pas à des machines différentes, dès lors que le niveau d'abstraction a permis de se sortir de l'adaptation à une machine particulière, mais à des humains différents, qui construisent le nouveau langage avec leurs logiques propres

⁵⁴ Ce qui fait dire à l'auteur que certains choix dans la logique du langage sont ses choix, ceux qui lui réservent le moins de surprise, voir l'URL : <http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-talk/143691> (consulté le 13 avril 2019).

(individuelle, ingénieriale, industrielle, commerciale). Toutefois, la possibilité d'extension de l'usage du langage est soumise à son niveau de standardisation, qui permet son interopérabilité. Ainsi, les langages de balisage léger, derniers arrivés parmi les langages de description, apparaissent pour répondre à une activité d'écriture limitée de leur auteur. Ce sont d'abord des *idiolectes numériques*, qui se situent au croisement du contexte machinique et du contexte social dans lesquels ils se déploient. En tant que langage informatique, ils doivent un tribut de compatibilité aux machines standardisées (celles de l'ère du *personal computing*) et aux langages disponibles. En tant que langages sociaux, ils répondent à des besoins humains spécifiques, ce qui conduit forcément, dès que leur adoption dépasse le premier cercle des adeptes précoces, à une interrogation de leurs limites premières.

3.3. Une norme et ses variantes

3.3.1. L'héritage du HTML

Ainsi que le remarque Antoine Fauchié, « *l'objectif initial de la création d'HTML est double : structurer des contenus, [...] et permettre une interopérabilité. Ce dernier point est essentiel pour comprendre le succès d'HTML* » (Fauchié, 2017). D'autre part, l'interopérabilité est aussi au fondement de la simplicité du HTML en tant que langage informatique.

« HTML est intentionnellement conçu pour ne pas être un langage de programmation complet, de sorte que beaucoup de choses différentes peuvent être faites avec un document HTML : le logiciel peut présenter le document dans différents styles, extraire les tables des matières, l'indexer, etc. » (Berners-Lee, Mendelsohn, 2006, Nt)

Le Markdown hérite ainsi du HTML sa neutralité stylistique et sa simplicité langagière, qu'il réduit encore en choisissant de n'en simplifier qu'une petite partie : « *Markdown ne remplace pas le HTML, ni même s'en rapproche. Sa syntaxe est très petite, ne correspondant qu'à un très petit sous-ensemble de balises HTML.* » (Gruber, 2004d, Nt). De ce point de vue, il se conforme à la loi de moindre puissance telle qu'exposée par le W3C :

« Il existe un compromis important entre la puissance de calcul d'un langage et la capacité de déterminer ce que fait un programme dans ce langage. [...] De nos jours, nous devons comprendre les raisons de choisir non pas la solution la plus puissante, mais la moins puissante. L'expression de contraintes, de relations et d'instructions de traitement dans des langages moins puissants augmente la flexibilité avec laquelle les informations peuvent être réutilisées : moins le langage est puissant, plus vous pouvez en faire avec les données stockées dans ce langage. » (Berners-Lee, Mendelsohn, 2006, Nt)

Gruber revendique ce minimalisme, arguant des fondamentaux de simplicité du langage :

« Si Markdown fournissait sa propre syntaxe pour toutes les balises HTML que les gens pensent vouloir, ce serait un champ de mines de cas spéciaux dont vous auriez besoin d'être conscient à tout moment en écrivant. Une grande partie du succès (toujours croissant) de Markdown est que j'ai gardé la syntaxe relativement petite. » (Gruber, 2015, Nt)

Nous pensons que la popularité du *Markdown* est paradoxalement due à ces limites : incomplet en tant que langage informatique, il l'est aussi en tant que sociolecte numérique, mais cela favorise son appropriation en ouvrant les possibilités de sa modification ; le caractère *flexible* du langage facilite son adoption par une frange plus large de la communauté utilisatrice, quitte à ce que cela pose ensuite des problèmes d'interopérabilité.

3.3.2. De l'individu à la communauté, de la norme au standard

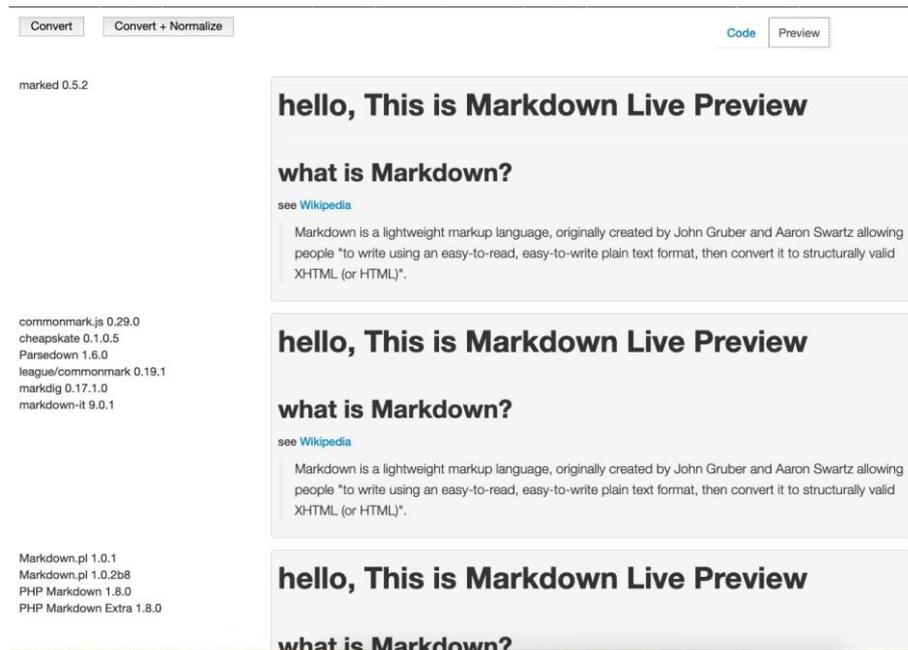
Si l'argument de la simplicité revient dans toutes les discussions sur les avantages du *Markdown*, à l'opposé, toutes les critiques convergent vers les difficultés inhérentes à sa non standardisation. Avec l'évolution exponentielle de la publication numérique, en ligne ou non, le *Markdown* devient populaire comme langage d'écriture et de publication, quelle que soit la spécialisation du rédacteur : les différentes variantes du *Markdown*, souvent nommées *flavor* (parfum, ou saveur), de manière assez suggestive, entérinent cette évolution. En l'absence d'une direction donnée par son fondateur, le *Markdown* ouvre la porte à une multitude de variantes qui l'éloignent autant de

l'interopérabilité, chacune ajoutant ses spécificités à la norme déjà bancaire de l'original. L'un des développeurs les plus avancés dans l'évolution du Markdown a même mis au point, en guise de test, une application permettant de mesurer les différences entre les 22 versions connues du langage, qu'il nomme **babelmark**⁵⁵, en référence à l'épisode biblique.

Figure 1. Babelmark2 en action : le code Markdown en haut, les premières différences de conversion en HTML en bas, avec le nom des variantes.

⁵⁵ Voir l'URL : <https://johnmacfarlane.net/babelmark2/#results>. La première version de Babelmark est écrite par Michel Fortin, le premier développeur à avoir implémenté *Markdown* en PHP. Elle a été reprise par John MacFarlane, qui en a fait une version décentralisée (voir l'URL : <https://johnmacfarlane.net/babelmark2/faq.html>), dont nous nous sommes servis pour l'illustration.

Figure 2. Même code Markdown, mais en montrant le résultat des prévisualisations. Notons que les variantes n'agissent pas de façon homogène entre la conversion HTML et la prévisualisation (le rendu) du code final.



Babelmark illustre bien la prolifération des variantes : le Markdown est devenu un sociolecte numérique de fait, voire un langage numérique véhiculaire. Sa notoriété est en effet d'abord acquise auprès de la communauté des développeurs, et assise après le choix de son implémentation sur **StackOverflow** (<https://stackoverflow.com>), le site de discussion de référence chez les développeurs, créé par Jeff Atwood, et sur **GitHub**, la plateforme de référence de dépôt de code. Cela donne à Atwood un poids certain dans l'évolution du langage. Après plus d'un an d'observation de l'implémentation du Markdown sur **StackOverflow**, dans un ambivalent billet de blog de 2009, où il dit à la fois son soutien au Markdown et son exaspération quant à l'inconséquence de son auteur, Atwood finit par poser la question de « *la responsabilité*

parentale du code open source »⁵⁶. Il n'est pas seul dans ses récriminations ; plusieurs développeurs importants des variantes les plus connues du Markdown discutent avec lui de l'opportunité de développer un standard pour le langage de balisage. Atwood lance un appel aux bonnes volontés dans un article de 2012 intitulé « L'avenir du Markdown »⁵⁷, dans lequel il :

« propose que Stack Exchange, GitHub, Meteor, Reddit et toute autre société générant beaucoup de trafic et investissant de manière stratégique dans Markdown travaillent ensemble pour mettre au point une spécification officielle de Markdown et des suites de tests standard permettant de valider les implémentations de Markdown » (Atwood, 2012).

Le travail de standardisation est en marche. Les enjeux économiques viennent ajouter leur poids aux enjeux numériques. En 2014, le *Standard Markdown* est présenté au public :

« Nous avons passé des mois à bricoler les détails et à convenir des changements nécessaires pour transformer Markdown en une langue que vous pouvez analyser sans avoir l'impression de marcher dans un égout, tout en préservant l'esprit simple, clair et inspiré de la messagerie ASCII. [...] Cela a pris un certain temps, mais je suis heureux d'annoncer que Standard Markdown est maintenant fin prêt pour la consultation publique. » (Atwood, 2014)

La critique du Markdown original est acerbe, et le nom de Gruber est évoqué une seule fois, dans une citation qui critique implicitement sa gestion du langage :

« Comme la description de la syntaxe de Gruber laisse de nombreux aspects indéterminés, écrire une spécification précise nécessite de prendre un grand nombre de décisions, dont certaines sont quelque peu arbitraires. En les faisant, j'ai fait appel aux conventions existantes et à des considérations de simplicité, de lisibilité, de pouvoir expressif et de cohérence. » (Idem.)

⁵⁶ Voir l'URL : <https://blog.codinghorror.com/responsible-open-source-code-parenting/> (consulté le 7 juillet 2019).

⁵⁷ Voir l'URL : <https://blog.codinghorror.com/the-future-of-Markdown/> (consulté le 7 juillet 2019).

Les principes pragmatiques idiolectaux de Gruber, consistant à « *faciliter la lecture, l'écriture et la révision de la prose* » (Gruber, 2004a, *op. cit.*), ont vécu. La standardisation implique une montée en généralité qui se lit même dans le vocabulaire employé pour décrire la spécification, et qui le dépossède de *son* langage. Ses ultimes résistances (il fait changer le nom du projet *Standard*, puis *Common Markdown* en *CommonMark*⁵⁸) n'ont plus d'effet sur le devenir du langage. Le groupe de travail, composé de John MacFarlane, David Greenspan, Vicent Marti, Neil Williams, Benjamin Dumke-von der Ehe, et Jeff Atwood, procède en 2015 à une demande officielle de standardisation auprès de l'IESG (Internet Engineering Steering Group), le groupe de pilotage de l'ingénierie Internet responsable du processus de normalisation d'Internet.

Le CommonMark est relativement récent, son adoption est encore faible (malgré le coup de pouce du GitHub Flavored Markdown, qui s'appuie sur lui), et la spécification n'est pas encore finalisée, comme l'indique le site *CommonMark* (<https://commonmark.org/>) :

« The current version of the CommonMark spec is complete, and quite robust after a year of public feedback ... but not quite final.

With your help, we plan to announce a finalized 1.0 spec and test suite in 2019. »

Mais si on ne peut encore mesurer les effets de la spécification, on peut déjà dire que le Markdown a quitté son statut d'idiolecte numérique pour devenir, d'une part, en tant que langage social, un langage véhiculaire par les variantes qu'il a suscité et l'extension de leurs usages, d'autre part, en tant que technolecte informatique, un standard normalisé.

⁵⁸ Voir à ce propos le billet de Jeff Atwood, « Standard Markdown is now Common Markdown », et ses commentaires, pas dupes, voir l'URL : <https://blog.codinghorror.com/standard-Markdown-is-now-common-Markdown/> (consulté le 7 juillet 2019).

4. Conclusion

Comme l'écrivent Bouchardon et Cailleau, il s'agit de « *penser le numérique comme notre nouveau milieu d'écriture et de lecture. Être attentif au milieu de l'écriture, c'est comprendre qu'il n'y a pas de milieu technique qui ne soit aussi un milieu social (une communauté de savoirs et de pouvoirs)* » (Bouchardon, Cailleau, 2018, p. 121).

Les choix qui ont présidé à l'avènement du Markdown et ceux qui ont conduit à son évolution montrent que les langages informatiques sont des milieux techniques et sociaux. Le langage créé par John Gruber change de fonction et de statut au fur et à mesure de son évolution : en tant que code, il est le lieu même du dialogue humain-machine, passant d'idiolecte à standard, perdant en souplesse ce qu'il gagne en interopérabilité, en individuation ce qu'il gagne en cohérence. Sa mise en œuvre dans des éditeurs de texte nous permettra de montrer au plus près la sémiotique en acte de ce dialogue, car nous pensons, comme Bouchardon et Cailleau, qu'« *il y a un fort enjeu à tenter de rendre "visible" et "lisible" notre milieu numérique dans ses différents aspects : théoriques (propriétés spécifiques du numérique), techniques (fonctions qui les matérialisent dans des outils d'écriture et de lecture) et sémiotiques (pratiques sociales incarnées)* » (Idem.).

Bibliographie

- Atwood Jeff, Responsible Open Source Code Parenting, *Coding Horror*, 29 décembre 2009, URL : <https://blog.codinghorror.com/responsible-open-source-code-parenting/>
- Berners-Lee Tim, Mendelsohn Noah, « The Rule of Least Power », W3C, TAG Finding, 23 February 2006, URL : <https://www.w3.org/2001/tag/doc/leastPower>
- Bouchardon Serge, Cailleau Isabelle, « Milieu numérique et "lettrés" du numérique ». *Le français aujourd'hui*, janvier 2018, n° 200), pp. 117-126.
- Cailleau Isabelle, Bouchardon Serge, Crozat Stéphane, Bourdeloie Hélène (2012). « Compétences et écritures numériques ordinaires ». *Recherches en Communication*, n° 34, Louvain, Université Catholique de Louvain, pp. 33-50. Voir l'URL : <https://www.researchgate.net/publication/270216580CailleauIBouchardonSCrozatSBourdeloieH2012CompetencesetecrituresnumeriquesordinairesR>

recherches en Communication n°34 Louvain Université Catholique de Louvain 33-50

- Crozat Stéphane, Bachimont Bruno, Cailleau Isabelle, Bouchardon Serge & Gaillard Ludovic (2011). « Éléments pour une théorie opérationnelle de l'écriture numérique ». *Document numérique*, vol. 14 (3), pp. 9-33, URL : <https://www.cairn.info/revue-document-numerique-2011-3-page-9.htm>
- Davallon Jean, Després-Lonnet Marie, Jeanneret Yves, Joëlle Le Marec, et Emmanuël Souchier (2013) [2003]. *Lire, écrire, récrire : Objets, signes et pratiques des médias informatisés*. Études et recherche. Paris, Éditions de la Bibliothèque publique d'information, URL : <http://books.openedition.org/bibpompidou/394>
- Goyer Samuel (2017), *De briques et de blocs. La fonction éditoriale des interfaces de programmation (API) web : entre science combinatoire et industrie du texte*, Thèse en Sciences de l'information et de la communication, Paris IV Sorbonne, 2017.
- Gruber John (2004a), « Dive into Markdown ». *DaringFireball*, 19 mars 2004, <https://daringfireball.net/2004/03/diveintomarkdown>
- Gruber John (2004b), Markdown 1.0, *DaringFireball*, 28 août 2004, URL : <https://daringfireball.net/2004/08/markdown10>
- Gruber John (2004c), « Markdown ». *DaringFireball*, 17 décembre 2004, <https://daringfireball.net/projects/Markdown/>
- Gruber John (2004d), « Markdown : Syntax ». *DaringFireball*, 17 décembre 2004, URL : <https://daringfireball.net/projects/Markdown/syntax#philosophy>
- Gruber John (2015), « Markdown, Strikethrough, and Slack ». *DaringFireball*, 5 novembre 2015, URL : <https://daringfireball.net/linked/2015/11/05/Markdown-strikethrough-slack>
- Jeanneret Yves, Souchier Emmanuël (2005), « L'énonciation éditoriale dans les écrits d'écran ». *Communication et langages*, n° 145, 3^e trimestre 2005. pp. 3-15.
- Pignier Nicole, Drouillat Benoît (2008), *Le webdesign. Sociale expérience des interfaces web*, Paris, Hermès-Lavoisier, coll. « Forme et sens ».
- Saemmer Alexandra, Nolwenn Tréhondart (2014), « Les figures du livre numérique "augmenté" au prisme d'une rhétorique de la réception ». *Études de communication*, n° 43, voir l'URL : <http://journals.openedition.org/edc/6036>
- Stallman Richard M. (2017), « Le livre comme alternative pour l'école contemporaine ? ». *Hermès, La Revue*, février 2017, n° 78, pp. 104-107.