

Stéphane Crozat

EA 2223 Costech – Université de technologie de Compiègne – Sorbonne Universités

De l'écriture qui veut imprimer à l'écriture qui veut programmer

Introduction

L'écriture numérique est d'abord graphique, elle procède du signe interprété par l'humain ; mais l'écriture numérique est aussi computationnelle, au sens de Bachimont (2007), c'est-à-dire qu'elle procède du calcul effectué par l'ordinateur. Cette dualité pose des questions à l'ingénieur qui conçoit les instruments d'écriture et de lecture. Quand se reposer sur le graphique, l'héritage culturel, les conditions de l'interprétation ? Quand se reposer sur le computationnel, le traitement, l'automatisation ? Comment articuler les deux ? Mais ces questions sont également posées au citoyen, car les outils que nous utilisons pour lire et écrire instrumentent notre communication et notre pensée, comme l'a montré Goody (1977). Nous avons un intérêt direct à chercher à les comprendre en tant qu'instruments, pour s'en émanciper, éventuellement les maîtriser, au moins n'en être pas totalement esclaves. C'est l'enjeu du développement d'une littératie numérique.

L'écriture numérique : une inscription intentionnelle codée avec des logiciels via des ordinateurs

Notion d'écriture

L'écriture désigne originellement la représentation graphique d'une langue, par la spatialisation sur un support pérenne des sons ou des mots sous la forme de signes. Deux notions sont donc constitutives de l'écriture : la notion de langue qui fixe l'intentionnalité d'exprimer un contenu ; la notion de pérennité du support qui sous-tend la transmission différée de ce contenu. *L'écriture est un enregistrement intentionnel de signes sur un support pérenne en vue de transmettre un contenu.*

Initialement réservé à l'écriture alphabétique et idéographique, le terme a depuis été étendu à d'autres formes d'expressions : écriture cinématographique, radiophonique, voire photographique. Elle devient *écriture multimédia* dès lors que le support permet de juxtaposer différentes formes sémiotiques, et que la question de l'articulation entre ces différentes formes – l'intersémiotisation – se pose.

On pourra en corollaire définir les concepts associés de trace, inscription et document. Une *trace* est un enregistrement (et non une écriture) non intentionnel de signes (par exemple, une caméra de surveillance enregistre des traces). On entend ici non intentionnel, car s'il y a intention de faire trace, il n'y en a pas de produire une trace en particulier pour un objectif particulier. Une *inscription* est le résultat d'une écriture ; une inscription est donc produite intentionnellement en vue de tenir un propos (par exemple, une vidéo personnelle de vacances cadrée, montée...). Un *document* est une inscription établie dans un contexte éditorial (par exemple, un film commercial distribué sur DVD).

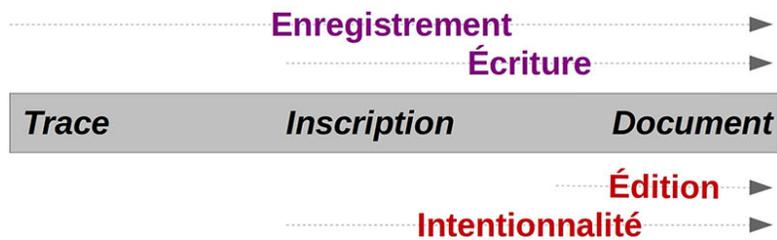


Figure 1 • Écriture = trace + intentionnalité

Écriture numérique

Le concept d'écriture numérique renvoie en fait à trois composantes du numérique :

1. L'écriture numérique est ce qui est produit avec un dispositif numérique de production des signes (un ordinateur et un logiciel d'écriture).
2. L'écriture numérique est ce qui produit des inscriptions qui sont numériques (un code binaire).
3. L'écriture numérique est ce qui produit des signes destinés à être interprétés via un dispositif numérique de lecture (un ordinateur et un logiciel de lecture et de d'interaction).

Écrire avec un logiciel

Toute écriture numérique procède de l'utilisation d'un ordinateur et d'un logiciel d'écriture. Ce dispositif peut tendre à masquer sa dimension numérique, pour simuler un dispositif graphique, comme une machine à écrire, une photocopieuse, une caméra ; ou bien intégrer sa dimension numérique en proposant explicitement des fonctions de calcul : copier/coller, chercher/remplacer, génération automatique d'index...

Écrire du code

Toute écriture numérique conduit à la production de code qui représente le contenu sous forme de chiffres. Cette inscription numérique pourrait n'être envisagée que comme intermédiaire, si la visée de l'écriture restait la production exclusive et *one shot* (d'un trait) d'un support matériel analogique, par exemple une impression sur papier. Mais il faudrait, pour se rapprocher de cette situation idéale, qu'il n'y ait pas de conservation de la production ni au cours de l'écriture, ni pour une révision ultérieure (ce qui n'est jamais envisagé en pratique). En effet, dès lors qu'il y a un enregistrement du code, il y a une gestion informatisée de ce code (j'entends ici enregistrement au sens de stockage dans mémoire secondaire, l'inscription est nécessairement en mémoire primaire, mais l'on peut considérer que son caractère temporaire ne procède pas d'un enregistrement). Donc de fait, dans la majorité des cas, l'inscription numérique aura une vie en tant que telle, elle sera gérée (diffusé, copié, manipulée...) en tant que code par les utilisateurs et les systèmes.

Écrire pour l'ordinateur

Enfin, si l'écriture peut se confiner à la réalisation d'un support graphique imprimé sur papier, elle tend de plus en plus à viser la lecture sur écran, c'est-à-dire la lecture via l'usage d'un ordinateur, et donc de ses périphériques d'entrée (souris, clavier, écran...) et de sortie (écran, haut-parleurs...). Cette écriture marque une troisième rupture en intégrant dans l'acte de lecture des fonctions du numérique : la manipulation du texte via la souris (ou l'écran tactile) et une iconographie, la scénarisation non linéaire, le multimédia, l'extension du texte par les commentaires...

L'écriture numérique est une programmation informatique

L'écriture consiste donc en l'inscription intentionnelle de signes en vue d'une transmission et d'une interprétation par un être humain. L'écriture numérique se caractérise par le fait que les inscriptions sont codées via un dispositif d'écriture numérique et décodées via un dispositif de lecture numérique. Le résultat de l'acte d'écriture avec un ordinateur est un code informatique dont l'exécution produit des signes sur un terminal de lecture qui donnent lieu à des interprétations.

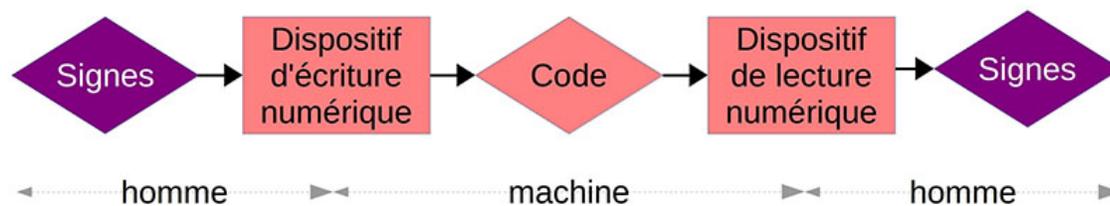


Figure 2 • Écriture numérique, entre signes et code

L'écriture numérique et la programmation informatique semblent *a priori* relever de deux pratiques différentes : l'écriture numérique consisterait à coder des données, les signes que l'on code et que l'on souhaite voir restitués « tels quels » ; tandis que la programmation consisterait à coder les calculs qui manipulent ces signes, ce qui serait de l'ordre du traitement.

Or cette séparation est fragile, d'abord au niveau logique de l'informatique où la frontière entre données et traitements est poreuse ; mais également au niveau sémiotique de l'écriture où la frontière entre écriture de contenus et écriture de programmes est également difficile à établir systématiquement dans le contexte du numérique.

Ce qu'il reste de la séparation entre écriture et programmation se cristallise dans le degré de conscience et d'intentionnalité de programmer que l'acte d'écrire inclut. Écrire relève d'une programmation, mais celle-ci peut être plus ou moins prégnante sur celle-là ; et la maîtrise de l'articulation entre le niveau logique de la programmation et le niveau sémiotique de l'écriture est un enjeu majeur de l'écriture numérique.

Écriture numérique traditionnelle et écriture numérique computationnelle

Toute écriture numérique procède donc d'une programmation par nécessité, écrire sur support numérique c'est programmer l'apparition de signes sur un support de lecture. En revanche, l'écriture peut procéder ou non d'une programmation par *intention*, c'est-à-dire exploiter ou non la logique du calcul au niveau sémiotique.

Lorsqu'elle adresse un support statique, comme le papier imprimé, l'écriture est graphique en intention et reste inscrite dans le paradigme traditionnel. Dès lors qu'elle adresse un support dynamique, tel qu'un écran d'ordinateur, l'écriture peut alors s'inscrire dans un autre paradigme, que j'appellerai computationnel, en ce sens qu'il est fondé par l'ordinateur comme support d'inscription, et qu'il relève de la raison computationnelle, c'est-à-dire d'une façon de penser avec le calcul que permet l'ordinateur (Bachimont, 2007). *L'écriture computationnelle a pour intention de programmer dynamiquement l'apparition des signes par l'exécution d'algorithmes et en fonction d'actions du lecteur.*

Les questions sont alors de savoir pourquoi et comment conserver une écriture traditionnelle, et pourquoi et comment mobiliser une écriture computationnelle. Questions qui se déclinent notamment au travers de l'instrumentation des logiciels d'écriture qui renvoient plutôt au paradigme du graphique (WYSIWYG) ou au paradigme du computationnel (WYSIWYM).

L'écriture qui veut imprimer

Le paradigme traditionnel

Une écriture numérique est traditionnelle si elle cherche d'abord à préserver les propriétés de l'écriture graphique, si elle n'est numérique que par obligation, happée malgré elle par l'universalité du numérique. Son enjeu est d'abord d'être capable de se reposer sur les compétences – d'écriture et de lecture – acquises dans le contexte du graphique. L'écriture numérique traditionnelle va nécessairement profiter de certaines fonctions spécifiques, mais de façon limitée, et essentiellement orientées vers le stockage et la circulation du contenu, plutôt que vers la production et la mise en forme des signes eux-mêmes. Cette résistance aux fonctions du numérique est essentielle dans la mesure où c'est la condition pour faire l'économie de la réinterrogation des modes d'interprétation. Elle doit tellement ressembler à l'écriture graphique non numérique que les mécanismes d'interprétation semblent fonctionner sans solution de continuité. On peut citer comme exemple l'écriture de documents PDF destinés à être imprimés, ou de document Epub à destination de liseuses passives (c'est-à-dire non ou très faiblement interactives).

Pourquoi conserver l'écriture traditionnelle ?

Conserver la possibilité d'une écriture traditionnelle sur support numérique permet de s'inscrire dans les habitudes établies de ce que l'on a appris à l'école, de ce que l'on sait avoir du sens. L'écriture traditionnelle est *facile*, c'est ce qui nous paraît naturel, puisque c'est *déjà* acquis. L'écriture traditionnelle est *fiable*, c'est un mode de communication codifié *a priori* que nous maîtrisons, pour lequel notre potentiel de contrôle, de prédiction, est important. L'enjeu principal est de fait celui de l'efficacité : l'écriture traditionnelle, est une écriture qui fonctionne d'emblée.

Problématique de l'écriture traditionnelle

Le risque principal de l'écriture traditionnelle n'est pas sa disparition, la forme livresque (graphique, linéaire, non interactive) est un mode de communication de l'information puissant en soi (même si cette forme passe d'un statut très largement majoritaire à un statut minoritaire). Le risque principal est plutôt de ne pas assumer ou de mal en contrôler les limites et de conduire à de mauvais hybrides, qui se revendiqueraient encore du graphique, tout en ayant trop transigé avec les fonctions numériques. Les conditions interprétatives du graphique ne s'appliqueraient déjà plus, sans que l'on s'en rende compte, entraînant une perte de sens. Cette « sortie » involontaire du graphique peut être due à une pression mal contrôlée de la technique, il y a donc une nécessité de lutter en permanence contre les tendances du numérique. Elle peut également être due à un défaut d'analyse, de celui qui chercherait les avantages du numérique sans en accepter les transformations, pensant que l'écriture n'a pas besoin d'être repensée, que le passage par l'ordinateur ne change rien, que « l'ordinateur n'est qu'un outil ».

L'écriture qui veut programmer

Le paradigme computationnel

L'écriture numérique est computationnelle lorsqu'elle cherche d'abord à profiter des fonctions du numérique, que ce soit pour améliorer la productivité (de l'écriture ou de la lecture) ou pour élaborer des formes d'expression différentes de celles disponibles dans le monde graphique. Deux écritures s'opposent donc, dont la frontière serait plutôt simple à tracer *a priori* : toute écriture qui sort du graphique, du linéaire, du non interactif, tomberait de fait dans le computationnel. Ce serait la condition pour que l'écriture traditionnelle puisse se revendiquer de l'héritage quasiment intact de l'écriture graphique. Sous l'influence des tropismes du numérique – de ce qu'ils suscitent en termes d'interactivité, de multimédia, d'archivage, de connexion, de diffusion –, on observe un développement de formes multiples d'écritures computationnelles. Nous avons formalisé ailleurs la notion de tropismes du numérique (Crozat, 2015), dont je donne un aperçu en annexe 3.

Pourquoi mobiliser une écriture computationnelle ?

La première raison de s'inscrire dans l'écriture computationnelle est peut-être qu'il est difficile de faire autrement, sous la pression des tropismes du numérique donc, qui se manifestent sous plusieurs formes.

Une *pression technologique* fait évoluer les supports de restitution – l'exemple des dispositifs portables, *smartphones* et tablettes, est particulièrement prégnant – et ces nouveaux supports peuvent imposer de fait de nouvelles logiques de lecture. Par ailleurs, les modes d'accès en réseau font également pression sur les modes d'écriture ; l'écriture collaborative synchrone ou asynchrone étant un exemple significatif. Or pour écrire à plusieurs, en réseau, il est nécessaire de mobiliser la dimension computationnelle de l'écriture, pour échanger des commentaires, comparer des versions...

Une *pression organisationnelle* liée à la numérisation des métiers conduit également petit à petit à s'inscrire dans une écriture computationnelle. On pourra citer comme exemple la production des métadonnées et la mise en place de *workflows* d'écriture, la nécessité de respecter des règles opérationnelles d'interopérabilité ou légales d'accessibilité.

Une *pression personnelle* se manifeste également au niveau individuel : l'envie de bien faire ou de faire comme les autres et le sentiment de pouvoir faire « mieux » avec le numérique.

Problématique de l'écriture computationnelle

Le premier problème posé par l'écriture computationnelle est qu'il remet en cause les processus interprétatifs établis. Ces nouveaux processus n'ont pas été, pour l'essentiel, appris à l'école (légitimement restée ancrée dans la sécurité de la tradition graphique), ils doivent donc être inventés par et pour ceux qui écrivent, et par et pour ceux qui lisent. Les canons de l'écriture graphique s'ancrent dans une tradition qui se compte en siècles (encyclopédies, essais, romans, journaux, ouvrages techniques, documentations, articles scientifiques...), tandis que ceux de l'écriture computationnelle sont en invention, et se comptent encore en années, dans les meilleurs des cas.

Par ailleurs, cette écriture est multiple et complexe et chaque composante (interaction, multimédia, commentaire...) fait appel à des dimensions interprétatives différentes. Il y a deux axes qui résonnent entre eux et se combinent pour former une matrice des possibles : un axe des fonctions d'écriture qui portent le champ du possible et un axe des types documentaires – des genres – qui définissent des structures générales d'écriture qui s'établissent par rapport à la poursuite d'une finalité (Crozat *et al.*, 2012).

Instrumentation des écritures : du WYSIWYG et du WYSIWYM

La plupart des systèmes d'écriture numérique modernes se fondent sur le principe du WYSIWYG, « *What you see is what you get* », ou littéralement en français : « *Ce que vous voyez est ce que vous obtenez* ». Cette approche vise, en résumé, à permettre à l'utilisateur de créer un document tout en composant le *rendu final*, comme dans sur un support traditionnel non numérique. Les premiers logiciels WYSIWYG datent des années 1970, mais leur démocratisation est associée aux premiers Macintosh en 1984 et aux traitements de texte MacWrite (1984) puis Word (1989).

Mais avant l'avènement des interfaces graphiques et du WYSIWYG, l'écriture numérique, essentiellement textuelle, procède de l'association des caractères à afficher avec des codes de formatage qui décrivaient *comment* les afficher (police, taille, couleur...). Par exemple, sous WordStar, les mots en gras doivent être placés entre deux balises « ^B ».

Cette approche originelle s'est prolongée à travers le principe WYSIWYM, « *What you see is what you mean* », c'est-à-dire : « *Ce que vous voyez est ce que vous voulez dire* », mobilisée dans les éditeurs LaTeX (LyX), HTML (WYMeditor) et les chaînes éditoriales XML (Scenari). L'idée est de continuer d'associer explicitement des codes aux contenus à afficher, mais, d'une part, que ces codes reflètent une intention auctoriale plutôt qu'un ordre de formatage, et, d'autre part, de mobiliser une mise en forme graphique adaptée au processus d'écriture (qui n'est pas le rendu final).

L'annexe I présente des copies des interfaces des outils mentionnés dans cette section.

Le WYSIWYG

L'instrumentation d'une écriture traditionnelle doit avoir pour objectif premier de masquer la dimension computationnelle de l'écriture. L'auteur qui écrit n'a pas besoin de savoir qu'il programme, puisqu'il ne *veut pas* programmer ; il est donc plus efficace de le maintenir dans un environnement qui simule l'ordre graphique qu'il maîtrise.

C'est sur cette hypothèse que reposent les fondamentaux du WYSIWYG : l'ordinateur simule une page blanche sur l'écran qui permet de se projeter directement dans l'imprimé qui résultera de l'acte d'écriture (comme le propose la traduction du terme WYSIWYG en français par « *tel écran, tel écrit* »). Les actes de programmation sont réalisés via des manipulations sémiotiques (cliquer sur une icône avec la souris) et provoquent immédiatement une manifestation sémiotique à l'écran qui préfigure un résultat équivalent sur l'imprimé (mettre un mot en italique).

Le WYSIWYM

L'approche WYSIWYM vise à proposer à l'auteur une interface de création de contenus qui se détache de la mise en forme finale, contrairement au WYSIWYG, et intègre explicitement des actes de programmation. L'écriture WYSIWYM consiste, d'une part, à produire *directement* une partie des signes, comme dans une approche WYSIWYG, et, d'autre part, à *programmer* d'autres signes qui seront générés algorithmiquement par la machine, ainsi que les rendus et les comportements dynamiques de l'interface de lecture. L'auteur déclare à l'aide de l'environnement d'écriture ce qu'il souhaite obtenir afin que la machine se charge de calculer une forme qui corresponde à ses intentions. Par exemple, l'auteur ne met pas de mots en gras ou en italique, mais ils les marquent comme des mots importants ou étrangers. Il ne définit pas la police ou la numérotation des titres, mais il déclare la structure du document. Il n'utilise pas d'encadrés ou de couleurs, mais il spécifie des blocs fondamentaux, des définitions, des exemples, des compléments. C'est dans un second temps que le programme informatique transformera chaque marqueur en une mise en forme adaptée à la lecture, en fonction du support.

Le WYSIWYM permet d'instrumenter l'écriture computationnelle en préservant un compromis entre ouverture des possibilités calculatoires via une programmation déclarative de haut niveau et maintien d'une inscription graphique propre à l'écriture.

Étude de cas d'écriture computationnelle

Dans cette partie, nous présentons quelques cas d'écriture computationnelle typique. Ces cas sont illustrés au travers d'exemples issus de la chaîne éditoriale Scenari (scenari.org).

Un premier cas simple : insérer une image dans un texte

La plupart des écritures numériques mêlent le texte alphabétique et les images, l'universalité du support y conduisant rapidement. Les traitements de texte permettent ainsi simplement d'ajouter une image. Pour cet exemple, nous prendrons un texte sans importance au sein duquel nous insérerons des images issues de plans de Léonard de Vinci (Cianchi, 1984). La première image, *Étude de polyèdre*, contient peu de détail et aucun texte ; la seconde, *Four à réverbère*, contient du texte et des images détaillées ; tandis que la troisième, *Chaînes*, contient de nombreux détails, mais pas de texte (voir annexe 2).

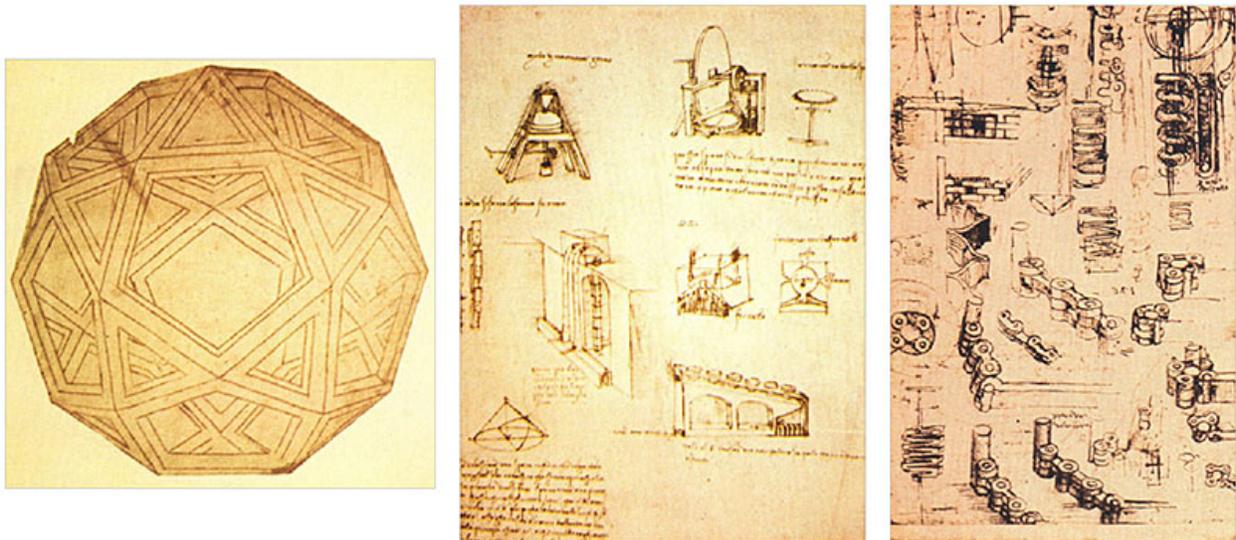


Figure 3 • Étude de polyèdre, Four à réverbère, Chaînes (Léonard de Vinci)

Traitement de texte WYSIWYG (écriture traditionnelle)

Dans un traitement texte traditionnel, l'insertion de l'image se fera graphiquement de façon à obtenir le rendu désiré. On jouera sur les paramètres graphiques tels que :

- la taille de l'image (redimensionnement) ;
- le positionnement de l'image relativement au flux de caractères : avant, après, à droite, à gauche, entourée par... ;
- la rotation de l'image.

On voit sur les trois exemples qui suivent que l'approche WYSIWYG fonctionne parfaitement bien, tant que c'est une impression sur papier qui est visée.



Figure 4 • Mise en page orientée impression

Lecture sur écran

En revanche, si l'on projette ces trois exemples sur des supports dynamiques de lecture, de nombreuses critiques peuvent être formulées, par exemple :

- Pour le dessin Polyèdre : sur un petit écran (smartphone), le choix d'un vis-à-vis entre le texte et l'image n'est pas opportun, on préférera un bout à bout classique.
- Pour le dessin Four : sur un écran d'ordinateur, on pourra fournir une image très haute résolution permettant d'explorer les plus petits détails de cette image. Mais pour éviter un temps de chargement long ou un encombrement spatial de la page, on préférera sûrement afficher d'abord une réduction de l'image qui permettra d'activer l'image haute résolution sur une action de l'utilisateur. On pourra aussi dans ce cas ajouter une loupe permettant d'accéder à la lecture des notes manuscrites sur le schéma.
- Pour le dessin Chaîne : sur un écran d'ordinateur, le choix d'avoir pivoté l'image en mode paysage (qui a permis d'optimiser la mise en page sur papier) n'est plus adapté, l'écran ne pouvant se tourner aussi facilement qu'une feuille imprimée, d'une part, et la possibilité de défilement dynamique annulant la limite verticale de la page, d'autre part.

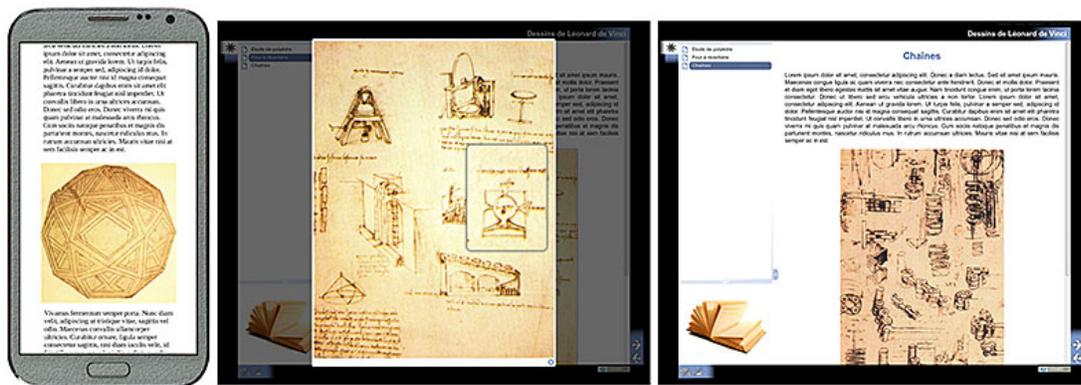


Figure 5 • Mises en page alternatives, dynamiques et orientées écrans

Chaîne éditoriale WYSIWYM (écriture computationnelle)

Ce que ces exemples nous montrent c'est, d'une part, que le polymorphisme et le WYSIWYG sont en contradiction intrinsèque, puisque, par définition, si ce que l'on voit est ce qu'on obtient, on ne peut voir, donc obtenir, qu'une seule configuration graphique, alors que chaque support de restitution requiert des configurations graphiques différentes (ici, la taille des images, leur alignement, leur orientation).

On voit également que l'écriture traditionnelle ne peut pas prendre en charge l'interaction du support de restitution. Il n'est en effet pas possible de systématiser la gestion interactive des images : disposer d'une miniature et d'une version haute résolution ou d'une loupe dépend du contenu de l'image (présence de détails, de caractères textuels) et de l'intention de l'auteur, et pas seulement de ses propriétés graphiques. Un auteur pourra décider que le dessin *Polyèdre* est destiné à une vue d'ensemble et donc qu'une fonction de zoom sera contre-productive, alors que rien ne permet de le détecter automatiquement. Une action explicite de paramétrage par l'auteur est donc nécessaire pour activer les bonnes fonctions de rendu et d'interaction.

Il y a alors passage à une écriture computationnelle, qui sait intégrer une part de programmation explicite dès lors qu'un écran est visé, et *a fortiori* plusieurs types d'écrans.

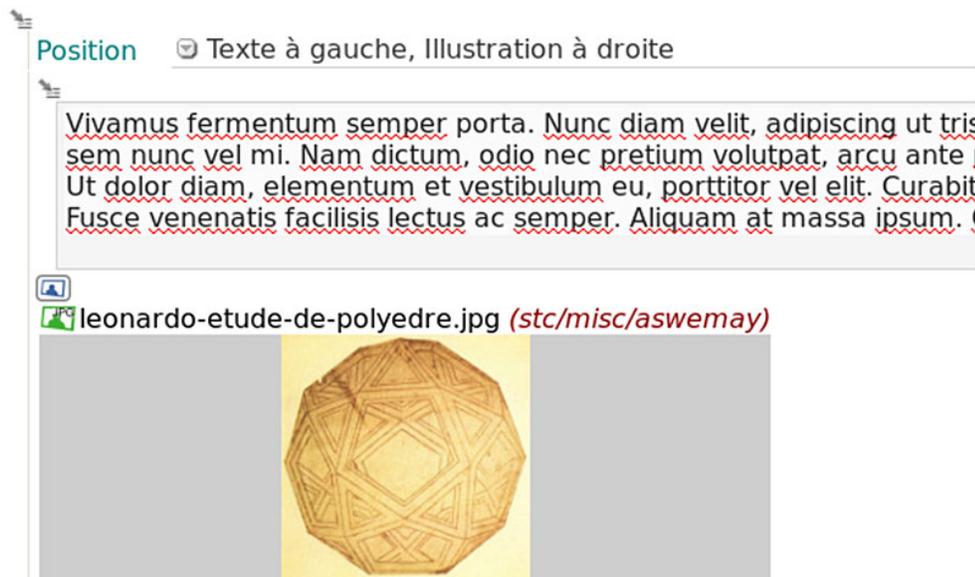


Figure 6 • Positionnement texte et image (Scenari)

Sur la figure 6, on visualise une saisie WYSIWYM du dessin *Polyèdre*. La déclaration de la position de l'image par rapport au texte n'est plus graphique, elle est programmée par l'auteur. L'auteur mobilise une fonction qui prend en entrée un texte, une image et un paramètre de positionnement ; cette fonction engendrera un comportement à la publication, comme par exemple : mettre en vis-à-vis sur un écran large, et mettre à la suite sinon. L'explicitation de la programmation ouvre un champ d'expression beaucoup plus large (d'autres fonctions, d'autres rendus) que ce que permettait le positionnement graphique WYSIWYG.

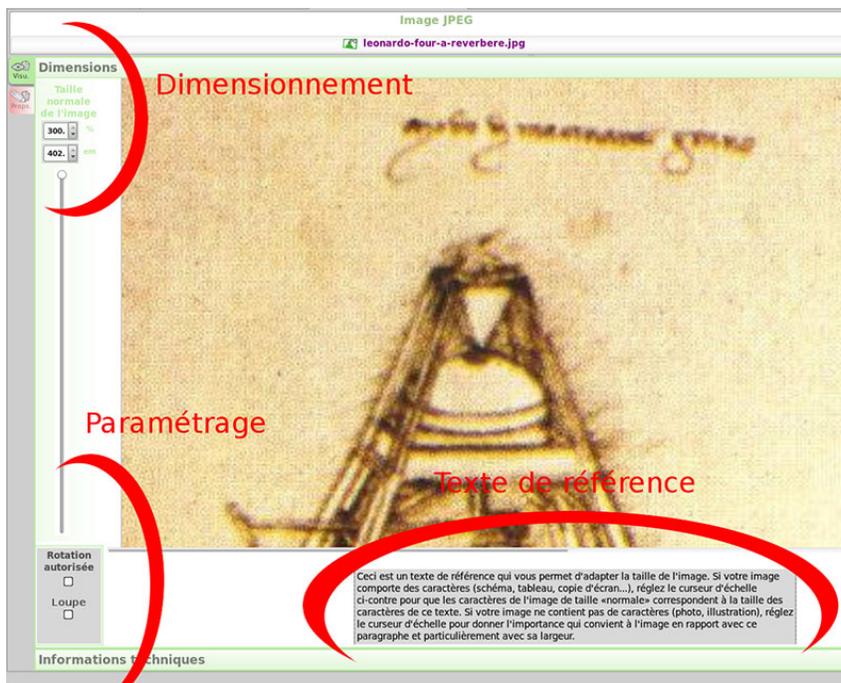


Figure 7 • Paramétrage d'une image (Scenari)

Sur la figure 7, on visualise le paramétrage de l'affichage de l'image correspondant au dessin *Four*. L'auteur en fixe une taille relativement à une taille de police standard d'un texte de référence. Ainsi, la taille de l'image n'est plus strictement graphique, exprimée en pixels ou en centimètres, elle devient un paramètre abstrait contrôlé par l'auteur, qui ainsi opère sur le calcul qui sera exécuté sur l'image à la publication. D'autres paramètres peuvent être mis à sa disposition, d'ordre graphique (la possibilité d'une rotation de l'image en cas de besoin d'optimisation d'espace de page) ou d'ordre interactif (l'activation d'une fonction de zoom ou de loupe).

Cette couche de programmation explicite apportée par le WYSIWYM permet à l'auteur d'élargir l'expression de son intentionnalité au-delà des seules caractéristiques graphiques de son écriture devenue computationnelle, pour étendre et exploiter l'algorithme qui contrôle le rendu dynamique du texte sur un écran d'ordinateur.

Autres cas

Écrire pour plusieurs supports en même temps

L'écriture numérique est de plus en plus une écriture multisupports, c'est-à-dire une écriture dont l'objet est de produire un contenu qui sera publié sur des supports de lecture différents : papier, écran d'ordinateur, écran de téléphone, vidéoprojection, tableau blanc interactif, liseuse... Afin que le contenu puisse être adapté à chaque support de restitution, l'auteur intègre des paramètres à certaines parties du contenu : pour spécifier qu'ils peuvent ou doivent être exclus de telle ou telle forme (ce paragraphe long est à exclure des formes synthétiques comme les diaporamas) ; pour proposer des formes alternatives à choisir en fonction des supports (cette série d'images sera proposée sur écran, mais sur papier on choisira cette seule image représentative).

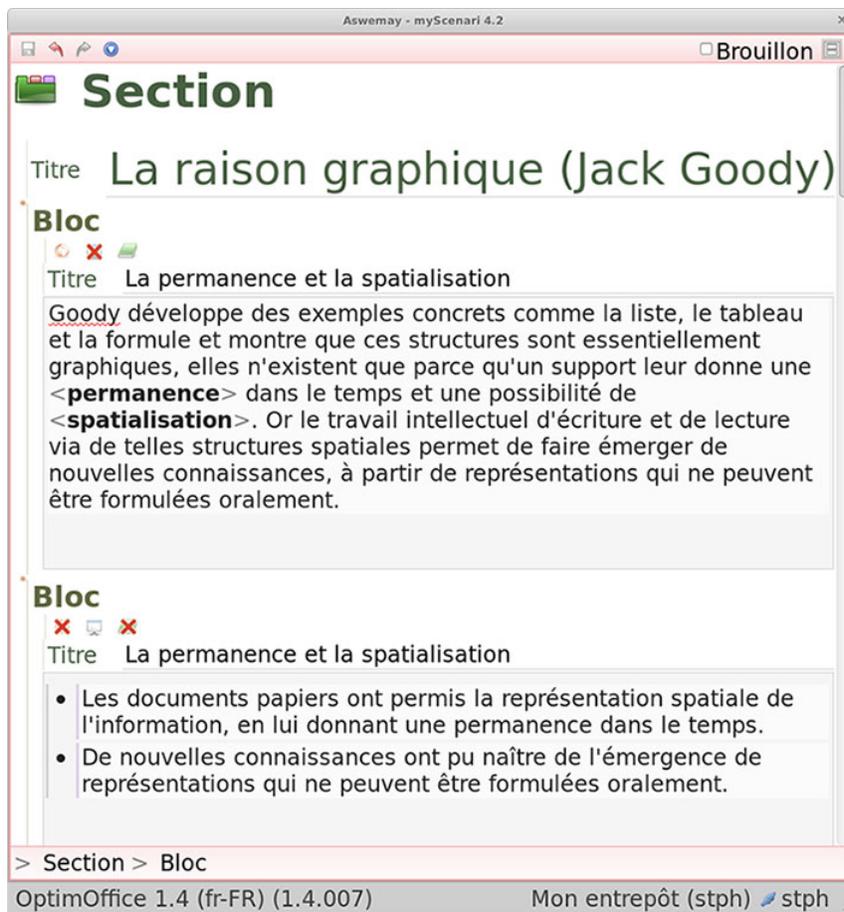


Figure 8 • Alternative entre une forme développée et une forme synthétique

Scénariser des parcours multiples

Dès les années 1970, Nelson (1974, 1981) pense le support numérique en rupture avec la linéarité héritée du livre en imaginant la fragmentation du texte, les liens hypertextes ou le texte étirable (*stretch-text*). Produire un contenu numérique implique de programmer les conditions d'activation des pages par le lecteur. La figure 9 présente un exemple d'écriture à profondeur variable : le premier bloc contient l'essentiel de l'information, qui sera présenté à tous les lecteurs ; le second est une illustration qui sera proposée à un second niveau d'information ; et enfin, les deux derniers blocs sont des compléments qui ne seront convoqués que sur demande explicite du lecteur. La figure 10 présente un exemple de programmation de parcours conditionnel dans un contenu de type « livre dont vous êtes le héros », l'accès à une page est ici conditionné à l'accès préalable à une autre page.

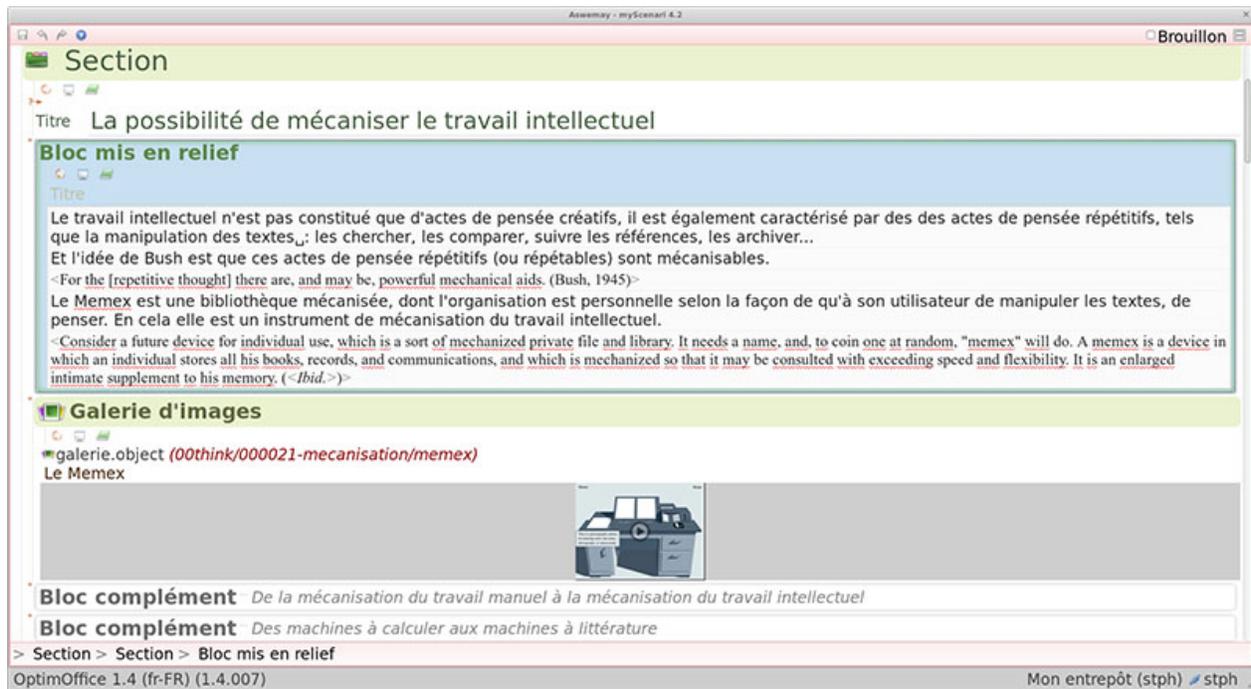


Figure 9 • Écriture à profondeur variable (stretch-text)



Figure 10 • Programmation d'un parcours conditionnel (si alors... sinon...)

Publier pour tout le monde (accessibilité)

Afin de répondre aux exigences d'accessibilité, il est nécessaire d'intégrer dans le processus de rédaction des éléments spécifiques comme les alternatives textuelles aux images, le sous-titrage des vidéos, les transcriptions de l'audio ou la description des liens hypertextes.

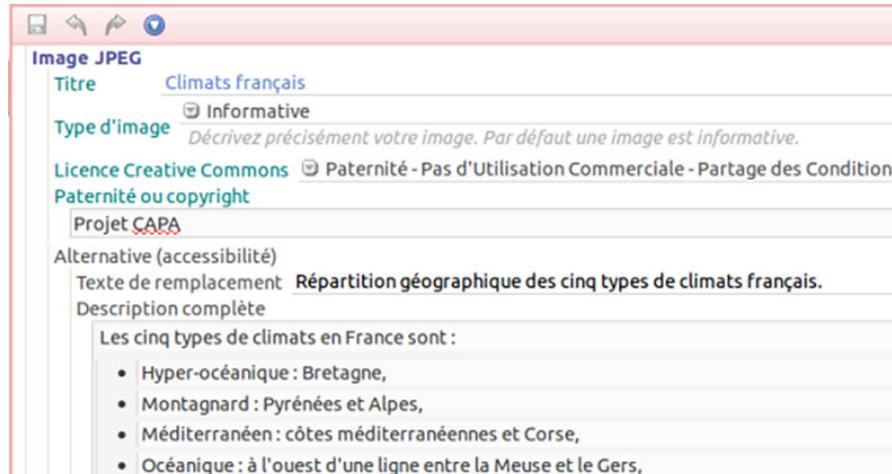


Figure 11 • Alternative textuelle à une image exploitable par un synthétiseur vocal (source : <http://ics.utc.fr/capa>)

Programmer des interactions

Le caractère computationnel de l'écriture se manifeste en contexte pédagogique à travers la programmation de l'interactivité des exercices et des quiz.

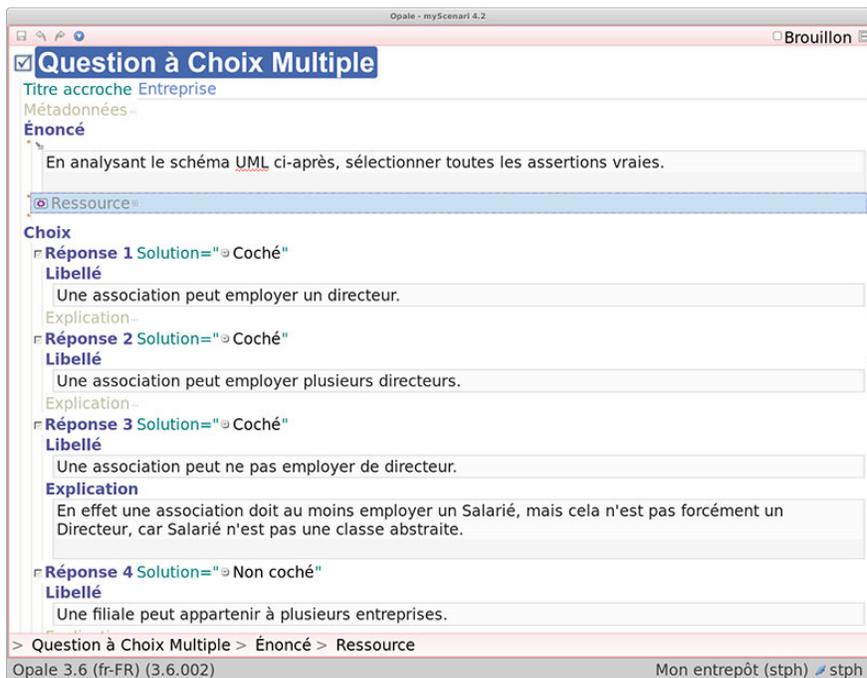


Figure 12 • Écriture et programmation d'une activité pédagogique de type quiz

Conclusion : notion de littératie numérique

Le dualisme écriture-programmation propre à l'écriture numérique repose la question de la littératie, c'est-à-dire des compétences de lecture et d'écriture qu'il est nécessaire de maîtriser pour manipuler l'information numérique.

Accéder au computationnel

La littératie numérique implique la possibilité d'appréhender le niveau calculatoire, le niveau sémiotique et l'articulation entre les deux. Il faut toujours la même appréhension au niveau sémiotique, mais il faut donc en plus une appréhension du niveau calculatoire qui sous-tend le niveau sémiotique que l'on perçoit ; et il faut une appréhension de l'articulation entre les deux niveaux. Cette nécessité d'*appréhension* porte la nécessité d'une pratique réfléchie pour devenir intuitive, accessible à la pensée pour être contrôlée. Il faut comprendre pour anticiper ce que l'on produit, la nature de ce que l'on peut faire et de ce que l'on ne peut pas faire, de ce que ça va devenir, des opérations que ça peut subir.

Un exemple

Prenons comme exemple l'écriture via un éditeur de texte simple en apparence comme Framapad, une instance d'Etherpad (éditeur collaboratif en temps réel) hébergée par Framasoft que l'on peut utiliser directement en ligne, sans identification préalable (*framapad.org*).

Au premier abord, l'écriture semble essentiellement graphique, orientée vers une production de signes alphabétiques, complétée par de la mise en forme basique (gras, italique). De fait, les documents produits avec Framapad sont simples. Et pourtant, si l'on y regarde de plus près :

- Le simple fait de saisir une adresse web produit un lien cliquable, sans qu'il soit besoin de programmer cette interaction (c'est le cas aujourd'hui dans la majorité des éditeurs de texte). Se pose donc d'emblée la question du bon usage de ces liens, problématique bien connue de l'écriture hypertextuelle.

- Framapad est un logiciel en réseau, ce qui est écrit est donc sur un serveur distant, ce qui impacte la disponibilité de ce contenu (il faut une connexion Internet et que le serveur soit fonctionnel) ou sa confidentialité (le contenu est public par défaut).

- C'est un logiciel collaboratif synchrone, donc d'autres utilisateurs peuvent modifier le contenu, y compris pendant que j'écris.

- Par ailleurs, tout ce qui est écrit dans Framapad est historisé automatiquement, et il est possible de rejouer la séquence d'écriture depuis le début, y compris ses fautes, ses hésitations, et bien sûr ces quelques mots géniaux que l'on avait cru perdus.

Ainsi donc, ce qui au premier abord pourrait sembler relever d'une écriture très traditionnelle, relève finalement largement aussi du computationnel. Cette analyse superficielle montre la nécessité de se questionner sur ce qui se passe au niveau logique du numérique. On notera que la superficialité de cette analyse fait ici son intérêt, elle montre que les fonctions computationnelles sont mal tapies, prêtes à surgir dès que l'on s'attarde un peu sur le chemin, facile à débusquer à qui les cherche un peu.

Littératie et technologie

Ce que je propose ici relève donc d'une démarche d'étude de la dimension technique des outils qui permettent de produire les textes, une démarche *technologique*, au sens étymologique. Pour bien écrire avec les outils numériques, ma thèse est qu'il faut comprendre le numérique dans sa

dimension informatique : les algorithmes, les variables, le stockage, la transmission, la recherche, l'identification, la modélisation... *Cela entraîne nécessairement une complexification et une abstraction des connaissances nécessaires à l'écriture, puisque l'on ajoute le niveau logique au niveau sémiotique, sans rien retrancher.*

Le choix des outils que nous concevons et utilisons pour écrire est donc fondamental parce qu'ils conditionnent notre écriture, ils sont prescripteurs de possibles. Le développement d'une littératie numérique nous permet de reprendre la main sur ces choix, de ne plus être simplement consommateur d'une informatique qui nous dépasse, de ne plus laisser uniquement le marketing décider de nos modes d'écriture à notre place. Etherpad ou Scenari, Powerpoint ou Writer, Gmail ou Thunderbird, Facebook ou Twitter, et l'indénombrable légion des autres logiciels qui existent ou qui existeront, peuvent être vus, doivent être vus, comme des logiciels d'écriture et comme des langages de programmation, dont il faut alors connaître les arcanes pour garder la maîtrise de son écriture. Mais comme il est impossible d'apprendre et de réapprendre en permanence les spécificités techniques et fonctionnelles de chaque logiciel, il est nécessaire de se doter de fondamentaux théoriques et techniques qui relèvent du numérique en général et sont donc transportables d'un logiciel à un autre.

Illettré numérique, on est relégué au rang d'utilisateur de services, on est prolétarisé au sens de Stiegler (2006), c'est-à-dire que l'on est privé d'un savoir et d'un savoir-faire que l'on possédait avant, que l'on ne peut plus décider de ce que l'on fait de notre écriture numérisée, de nos relations numérisées au monde. On est servi, et asservi, par des logiciels qui commandent à nos pratiques, des instruments qui nous instrumentent.

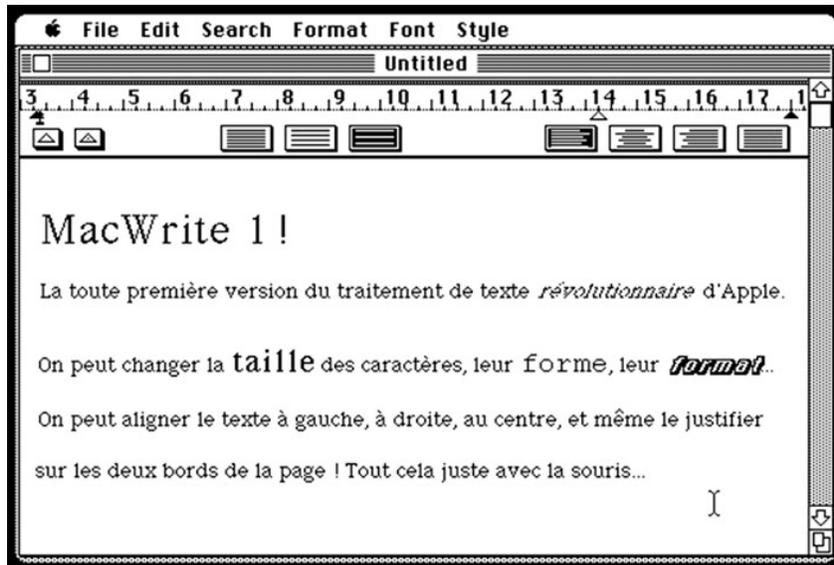
Une littératie numérique, au contraire, permet de comprendre les lois numériques qui gouvernent notre rapport au monde numérisé, comme les lois physiques gouvernent notre rapport au monde non numérisé. Elle permet de ne plus être le simple utilisateur de cette phrase érigée en bouclier : « *Je ne suis qu'un simple utilisateur.* »

Bibliographie

- Bruno BACHIMONT, *Ingénierie des connaissances et des contenus : le numérique entre ontologies et documents*, Paris, Lavoisier-Hermès, 2007.
- Marco CIANCHI, *Les machines de Léonard de Vinci*, Florence, Becocci Editore, 1984.
- Stéphane CROZAT, Bruno BACHIMONT, Isabelle CAILLEAU, Serge BOUCHARDON, et Ludovic GAILLARD, « Éléments pour une théorie opérationnelle de l'écriture numérique », *Document numérique* 2011/3 (Vol. 14), p. 9-33. Disponible en ligne : <http://www.cairn.info/revue-document-numerique-2011-3-page-9.htm>
- Stéphane CROZAT, « Les tropismes du numérique », in Imad Saleh et al. (dir.), *H2PTM'2015. Le numérique à l'ère de l'Internet des objets : de l'hypertexte à l'hyper-objet*, Paris, ISTE Éditions, collection « Systèmes d'information, web et société », 2015.
- Stéphane CROZAT et Sylvain SPINELLI, « Écriture computationnelle et passage au WYSIWYM : "What you see is what you mean" », in *CIDE 18 : Documents et dispositifs d'information à l'ère post-numérique*, Montpellier, 2015.
- Jack GOODY, *The Domestication of the Savage Mind*, Cambridge University Press, 1977.
- Theodor H. NELSON, *Computer Lib / Dream Machines* (1974), nouvelle édition, Microsoft Press, 1987.
- Theodor H. NELSON, *Literary Machines* (1981), nouvelle édition, Mindful Press, 1987.
- Bernard STIEGLER, *Réenchâter le monde : la valeur esprit contre le populisme industriel*, Flammarion, 2008.

Kees van DEEMTER et Richard POWER, « Authoring multimedia documents using WYSIWYM editing », in *Proceedings of the 18th conference on Computational linguistics*, vol. 1, p. 222-228, Association for Computational Linguistics, 2000. Disponible en ligne : <http://www.aclweb.org/anthology/C00-1033>

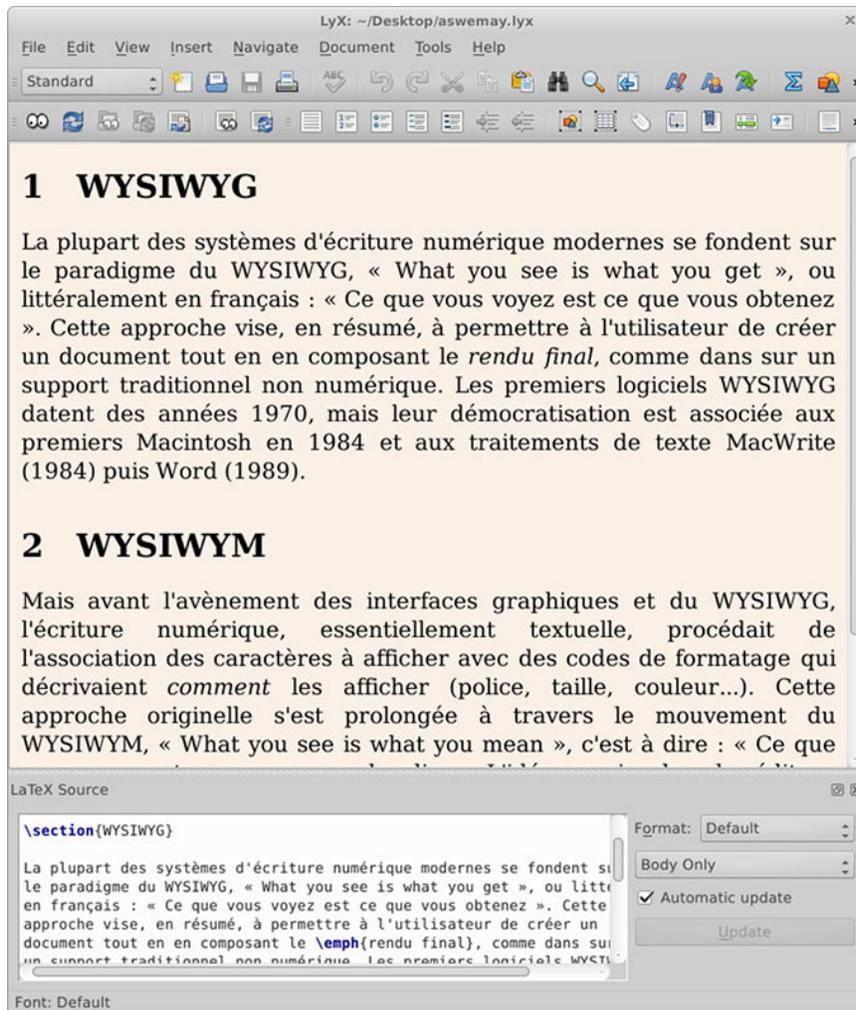
Annexe 1 – Outils WYSIWYG et WYSIWYM



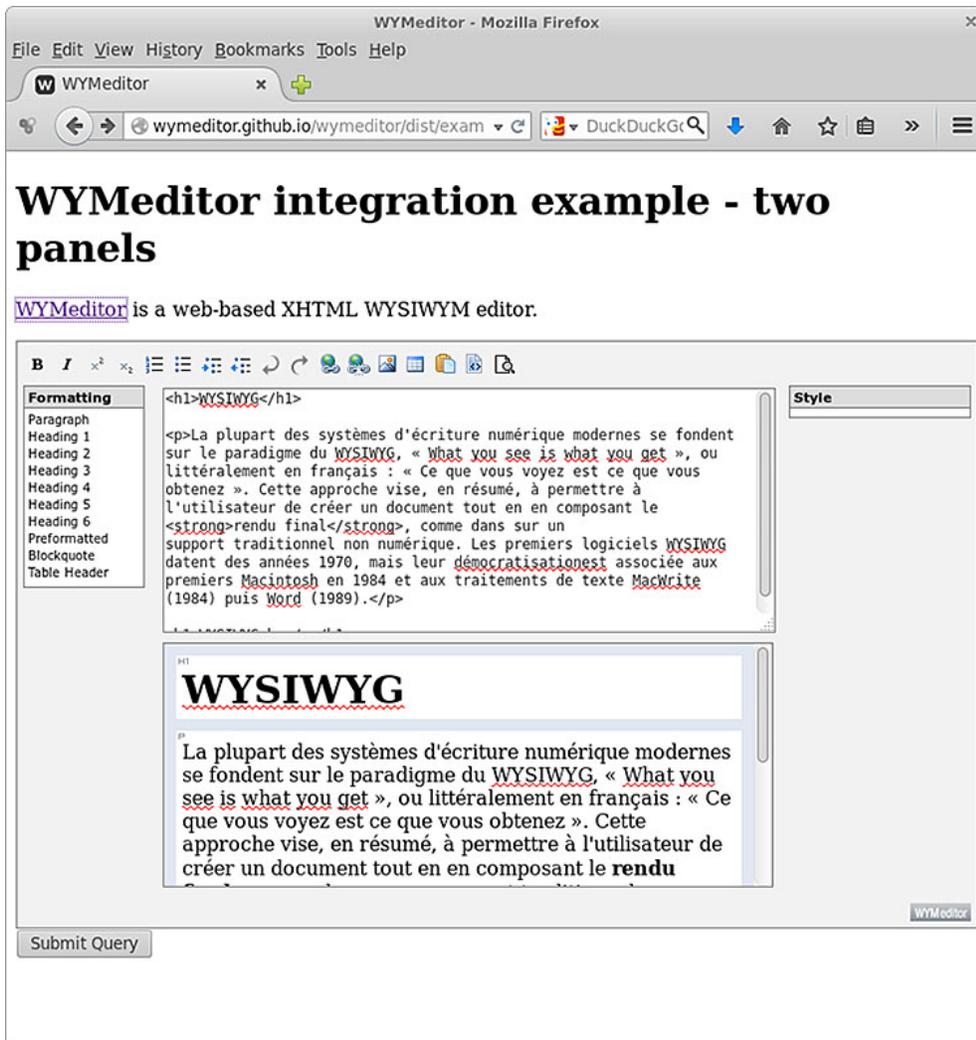
MacWrite, le premier traitement de texte WYSIWYG (source : <http://www.aventure-apple.com>)



WordStar, un traitement de texte avant le WYSIWYG (source : <https://en.wikipedia.org/wiki/WordStar>)



Lyx, premier éditeur WYSIWYM (lyx.org)

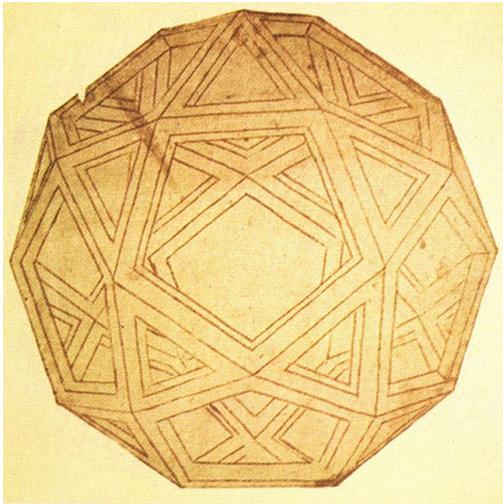


WYMeditor, éditeur Web WYSIWYM (wymeditor.org)

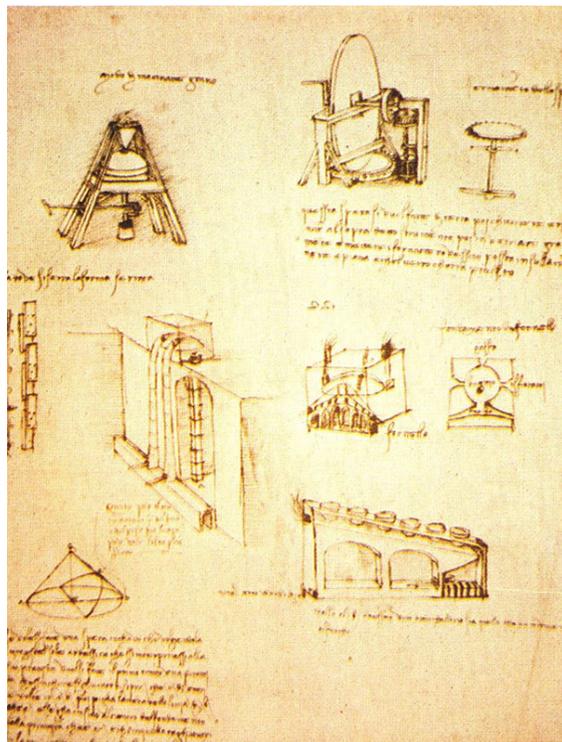


Scenari, éditeur WYSIWYM XML (scenari.org)

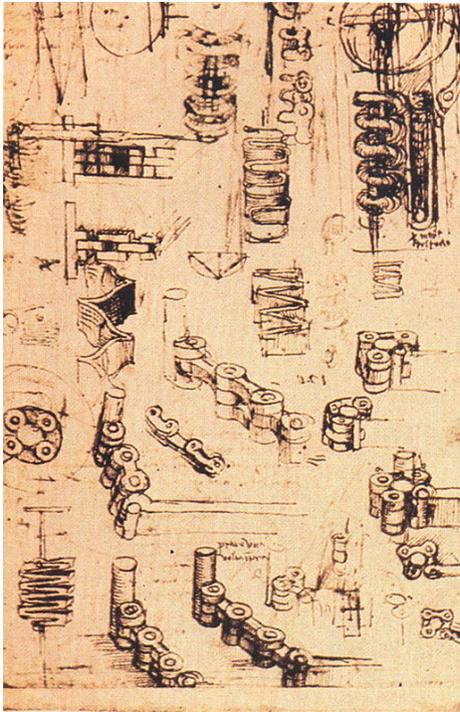
Annexe 2 – Trois dessins de Léonard de Vinci



Étude de polyèdre (Léonard de Vinci)

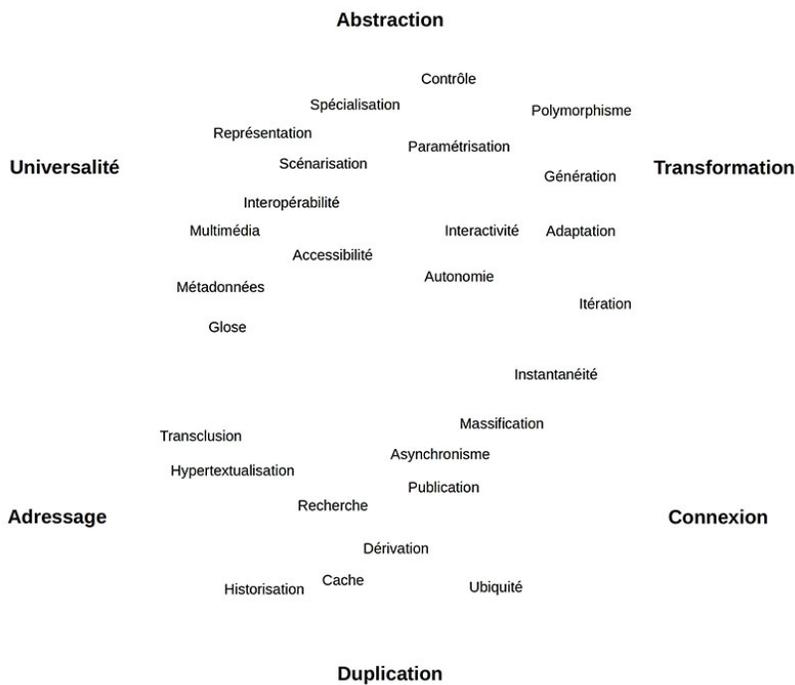


Four à réverbère (Léonard de Vinci)



Chaînes (Léonard de Vinci)

Annexe 3 – Carte des tropismes du numérique



<http://aswemay.fr/co/tropism-map.html>