

Un historique de la génération numérique de textes

Résumé

L'article se penche sur l'histoire de la génération numérique de texte définie comme la création d'un matériau linguistique par des moyens numériques et l'inscrit dans son contexte technico-culturel. Il démontre que la mise en relation entre le numérique et la textualité a lieu exactement à la même époque aux USA et en France, via le développement de l'informatique aux USA et celui de la robotique en France. Il remet en lumière les problématiques de la génération de texte : de technique dans un premier temps, la génération est devenue une question culturelle large puis littéraire. La première question traitée est celle de l'auteur, avant que la problématique ne bascule sur celle du texte, puis sur la lecture. Il résume les diverses conceptions qui se sont exprimées et soutient notamment que l'Oulipo n'a pas été un groupe moteur dans cette histoire. Enfin, il montre comment certaines démarches ont élargi le propos et mené à une rencontre entre génération de texte et animation pour déboucher, dans les années 2000, sur une approche plus générique de la poésie programmée.

Mots clés : littérature combinatoire, générateur de texte, historique

Abstract

The paper deals with the story of text generation defined as digital creation of a linguistic matter. It inscribes it in its techno-cultural context. It demonstrates that the relationship between textuality and the digital begins at the same time in the USA and in France but not on the same bases: with the development of computing science in the USA, and robotics in France. It highlights text generation problematics: being a technical issue first, text generation quickly

becomes a cultural and a literary issue. It first questions the nature of the author, and then the nature of the text and of the reading. It discusses different conceptions and argues that Oulipo has not been a driving force for digital text generation. Finally it talks about ways that have enlarged generation by a converging between text generation and text animation, leading to a programmed poetry approach.

Keywords: text generation, text combinatory, automatic generation, history of digital literature.

1. Introduction

Il faut se rendre à l'évidence : aucune définition de la littérature numérique n'a réussi à s'imposer. On peut alors se demander si les productions littéraires dites numériques relèvent bien d'une catégorie nouvelle, d'un nouveau genre littéraire, d'un processus dynamique de métamorphose des questions littéraires ou si la dénomination de « littérature numérique » n'est rien d'autre qu'un label marketing, un bon coup de com, en somme.

La dernière hypothèse ne saurait expliquer la structuration institutionnelle internationale de ce champ, l'existence d'enseignements, d'équipes universitaires, d'activités culturelles et de programmes de recherche parfois financés au niveau européen. Si la seconde hypothèse est la bonne, que nous sommes face à une métamorphose dynamique des questions littéraires, alors il n'est pas surprenant qu'on ne puisse définir dans l'absolu la littérature numérique, celle-ci évoluant jusque dans ses racines et ses conceptions au cours du processus de transformation. En revanche, il doit être possible de l'aborder d'un point de vue diachronique et d'en déterminer la nature en des lieux et des moments précis. Cette approche, la nôtre, déplace la question de la définition. Il ne s'agit plus d'aborder un « quoi », de répondre à la question « qu'est-ce que la littérature numérique ? », mais de questionner un « comment », « comment la littérature se fait-elle numérique ? » en un lieu et à une époque donnée. Il sera donc question des littératures numériques ou des formes de littérature numérique, et non de « La » littérature numérique. Ces formes ne peuvent naître et évoluer que dans un contexte techno-culturel donné, contexte qui évolue lui-même au cours du

temps. La prise en compte du contexte est essentielle pour comprendre pourquoi certaines conceptions et formes ne peuvent pas apparaître avant une date donnée et comment survivent les conceptions des périodes précédentes. Car c'est une constatation commune qu'en littérature une idée ou une forme n'en chasse pas une autre mais s'y superpose. Les formes naissent et meurent selon une courbe qualitativement en cloche définie par le contexte : pionnières au début dans un environnement qui ne les comprend pas (et leurs auteurs font parfois partie de cet environnement !), elles s'imposent ensuite comme formes standards et majoritaires avant de décliner comme survivance dans un contexte qui s'est focalisé sur une autre approche.

Je présenterai ici un point de vue sur l'évolution de la génération numérique de texte dans une optique française. Cette présentation serait sans doute différente aux États-Unis où s'est développé toute une pensée de l'hypertexte ou au Brésil, fortement marqué par la poésie concrète, ainsi, d'ailleurs que dans d'autres pays comme l'Allemagne où s'est également organisé un puissant débat littéraire.¹ Pour « globales » qu'elles sont, les formes littéraires numériques n'en sont pas moins fortement enracinées dans des cultures régionales ; la globalité est une circulation, non une uniformisation.

La succession des productions reconnues comme littéraires numériques est relativement bien connue, au moins jusqu'au développement du Web, notamment grâce au travail très documenté de Chris Funkhauser.² J'y décèle pour ma part quatre périodes. La première s'étend de la toute première production connue, celle de Christopher Strachey en 1952, à la fin des années 1970. C'est la période qui voit se développer les générateurs combinatoires de texte. La seconde période recouvre les années 1980-1990 et s'achève lentement à la fin des années 90 avec le développement de l'Internet. Cette période voit la question générative évoluer et être confrontée aux autres formes de base des productions littéraires numériques qui apparaissent à cette époque : l'animation de texte (1982),³ l'hypertexte de fiction (1987).⁴ On peut succinctement la caractériser comme un passage d'une littérature du livre à une littérature de l'écran. La troisième correspond aux écritures en ligne : écritures participatives puis blogosphère, écritures du flux sur la base de requêtes google, développement d'une interactivité non navigationnelle plus proche du jeu vidéo. On peut la caractériser comme le passage à une littérature « communautaire », c'est-à-dire fondée sur l'existence de communautés d'acteurs en interaction. Enfin, avec le développement depuis 2007⁵ environ des dispositifs mobiles et

des maisons d'édition en ligne, nous pourrions bien être dans la quatrième période caractérisée par un développement de productions programmées géolocalisées, l'utilisation de dispositifs interactifs évolués comme la kinect ou l'anémomètre des portables d'une part et d'autre part un transfert massif via le format epub vers des « supports » numériques d'oeuvres qui demeurent dans la tradition du livre imprimé, une période où se croisent les approches post-avant-gardistes centrées sur le dispositif et les approches plus classiques centrées sur la dimension linguistique du texte ; la coexistence d'une littérature numérique du dispositif et d'une littérature numérique de la structure. Il n'est pas possible, dans un chapitre d'ouvrage, de donner un aperçu complet de cette histoire et je renvoie le lecteur à d'autres ouvrages pour un panorama plus large.⁶ Aussi nous concentrerons-nous sur la génération de texte, classiquement définie comme la production informatique d'un matériau linguistique correct sur le plan syntaxique. Il s'agit d'une forme qui joue un rôle moteur en France dans les deux premières périodes. C'est même sous cette forme que débute la relation entre l'informatique et la littérature.

La relation entre la textualité et l'informatique se met en place dès les origines de l'informatique. Elle est, en premier lieu, le fait de scientifiques et non de littéraires. Bien évidemment, il n'est pas question à l'époque de penser une « littérature numérique », l'affirmation de l'existence d'une littérature spécifiquement numérique ne se fera qu'à partir des années 80 et ce n'est que postérieurement que les productions des années 1950 ont été considérées comme relevant déjà de la littérature numérique bien qu'elles diffèrent par bien des aspects des œuvres qui ont suivi.

Il nous faut donc en tout premier lieu rappeler le contexte scientifique et technologique qui a rendu possible ce type de productions.

2. La machine de Turing et ses limites

Il faut remonter à la machine de Turing (1936)⁷ pour voir apparaître une conception de l'informatique qui va largement contribuer à différencier le fonctionnement diachronique des œuvres numériques de celles conçues pour d'autres dispositifs, dont le livre. La machine de Turing n'est pas une machine réelle mais un modèle abstrait d'automate, bien qu'il soit possible de réaliser des machines réelles très proches du modèle, et même des machines totalement mécaniques.⁸ La machine de Turing n'est pas, en elle-même, une spécificité de

l'informatique. Son impact en informatique vient du fait qu'elle constitue tout à la fois un modèle générique sur lequel s'appuie tout langage de programmation et qu'elle a inspiré Von Neumann dans sa conception du modèle d'architecture qui prévaut encore dans nos ordinateurs actuels.

La machine de Turing se présente comme un ruban de taille infinie découpé en cases qui peuvent être vides ou dans lesquelles sont écrits des symboles d'un alphabet que la machine reconnaît. Une tête de lecture/écriture parcourt le ruban selon un programme dont chaque ligne de code L est une séquence de 2 types d'instructions : l'écriture et le déplacement élémentaire d'une case. Schématiquement, une ligne de code de la machine se présente donc de la façon suivante : $(D0, L, S0) \rightarrow (D1, S1)$ qui donne en français « si la tête de lecture lit la valeur $D0$ dans la case où elle se trouve et que la machine est dans l'état $S0$, alors elle accomplira la séquence L d'instructions dont le résultat sera d'écrire la donnée $D1$ sur le ruban, d'avoir déplacé la tête de lecture sur ce ruban et d'avoir modifié l'état interne de la machine qui est maintenant $S1$ ». Ainsi, une même donnée $D0$ lue pourra produire des résultats d'exécution différents pour chaque état interne $S0$ de la machine. L'ensemble des lignes de code constitue la « table » de la machine. On peut en effet écrire le programme sous forme d'un tableau dont chaque ligne est une ligne de code et dont les colonnes sont : donnée lue, état à la lecture, instruction, résultat produit, état produit

Je ne vais pas aborder ici l'ensemble des caractéristiques de ce modèle, mais mentionner celles qui auront un impact sur les écritures numériques littéraires qui suivront. Cet impact n'apparaîtra en fait qu'à partir des années 80 et concerne essentiellement les programmes d'animation. Il est lié au fait que le modèle de la machine de Turing est réellement implémenté techniquement et qu'il a induit, logiquement, une certaine conception du programme dont certains aspects vont à rebours du sens commun, créant ainsi un hiatus entre savoir technique et croyance (ou imaginaire) technique.

La première caractéristique du modèle est l'existence de l'état interne, véritable coup de génie de ce modèle. Or l'impact de cet état interne va à l'encontre de la logique commune : quand Monsieur tout le monde programme quelque chose, que ce soit en informatique ou dans n'importe quelle activité, il raisonne souvent de manière causale : « si j'ai telle donnée en entrée et que j'applique telle instruction, alors j'obtiendrai telle donnée au final ». Par exemple, un auteur qui programme le déplacement d'un objet à l'écran utilisera dans son programme source⁹ des lignes de code qui disent en gros « je lis la

position P de l'objet et je l'incrémente d'une certaine valeur X », ce qui, dans une formulation plus proche de celle utilisée dans le modèle de Turing, s'énonce : $(P, L) \rightarrow (P+X)$. L'auteur du programme néglige l'influence de l'état interne sur son instruction, il ne connaît en général même pas ce problème ; il s'attend à ce que ce résultat se produise sur toutes les machines, c'est-à-dire qu'il est indépendant de l'état interne de ces machines au moment de la lecture de la position et ne dépend que de ce qu'il a écrit dans le programme source. En fait, pour fonctionner correctement sur un ordinateur, le programme nécessite que la machine soit dans un certain état que, manifestement, l'auteur de ce type de programme source ne gère pas. Il pourra s'en suivre des déconvenues, pouvant aller jusqu'à des résultats totalement incompatibles avec la logique énoncée dans le programme source lorsque l'état interne réel de la machine n'est pas l'état implicite qui devrait exister au moment de l'exécution d'une instruction.¹⁰

La temporalité de la machine n'est pas pensée par Turing. C'est une seconde caractéristique de ce modèle. Il nous dit bien que la tête se déplace, mais pas comment gérer ses processus de déplacement et de traitement. Le modèle ne prend en charge que l'état initial du déplacement (« elle est ici ») et son état final (« elle est là ») mais le processus de déplacement lui-même est impensé (« à quelle vitesse de déplacement ? à quelle vitesse de lecture/écriture ? à quelle vitesse s'effectue le changement d'état ? est-il synchrone ou asynchrone à l'opération de lecture/écriture, ? »). Ce modèle ne traite que de la logique du programme (les procédures) et non de son fonctionnement physique (les processus). Cela va conduire à une conception très abstraite des productions numériques qui sera implicite en génération numérique de texte : « tout est dit dans le programme source », qui pourrait être formulée en littérature numérique par une expression du type « un texte généré est réductible à son algorithme », conception qui privilégie la dimension algorithmique de la littérature numérique au détriment de la dimension performative induite par l'exécution du programme. La plupart des auteurs ont, en fait, cette conception algorithmique qu'on peut énoncer selon le syllogisme : « j'ai programmé ce qui doit se passer, mon programme source décrit ce que j'observe sur mon écran, donc mon programme source produira toujours ce que j'observe sur mon écran ». Une autre formulation possible de cette conception algorithmique peut être « si je teste à l'instant t sur la machine m mon programme de sorte qu'il produise tous les résultats qu'il peut produire, alors j'obtiendrai le même jeu de résultats à l'instant t+infini sur n'importe quelle machine compatible avec la machine m ». La

conception algorithmique ne considère que la relation entre le résultat de l'exécution et l'énoncé du programme source, elle postule une relation causale entre le programme source et le résultat observé, à l'exclusion de toute autre cause. Elle privilégie les procédures du programme source pour prévoir le résultat observé alors que le modèle procédural que j'ai développé à partir des années 90 pose une relation de transformation entre le programme source et le résultat observé, il privilégie le processus d'exécution pour comprendre le résultat observé. Ainsi, la conception algorithmique est fondée sur une conception potentielle de l'informatique, le modèle procédural sur une conception virtuelle, les termes potentiel (au sens du possible) et virtuel étant à prendre dans le sens rappelé par Lévy.¹¹ Or il est apparu dès 1990 que la mise en forme des informations à l'écran, tous les processus observables, donc notamment l'animation, dépendaient fortement du contexte technique de l'exécution. Cette propriété fondamentale des dispositifs numériques est connue sous le terme de « labilité ». Elle est due aux solutions proposées par Von Neumann. Son architecture d'ordinateur est fondée sur le modèle de Turing mais, décrivant une machine réelle, elle doit bien sûr prendre en compte la dimension physique des processus. La temporalité va alors être gérée par une horloge interne et le ruban remplacé par une mémoire de stockage, la mémoire vive (RAM), reproduisant ainsi à une caractéristique fondamentale de la machine de Turing : son universalité, c'est-à-dire l'indépendance entre la table de la machine et l'ensemble ruban/tête. Cette dissociation entre software et hardware, totalement compatible avec le modèle de Turing, induit que la gestion de la temporalité, in fine, échappe au programmeur et échoit aux concepteurs du hardware, à travers les performances de ce dernier. Par ailleurs, l'évolution de la programmation : l'invention entre 1959 et 1963 des interruptions, du temps partagé et du système d'exploitation, segmente le programme exécuté en composants qui ne dépendent pas toutes du programme source. Tout se passe comme si le programme source ne contenait plus la totalité des lignes de code exécutées par la machine de Turing. De façon pragmatique, le programme source, pour son auteur, cesse d'être celui d'une machine de Turing.

3. La période combinatoire

a) La textualité numérique comme problématique scientifique

Il existe plusieurs définitions possibles de ce qu'est un ordinateur, mais l'une des plus couramment admise considère qu'un ordinateur est une machine construite selon l'architecture de Von Neumann¹² dans laquelle on peut dissocier le programme (« software ») des composants qui permettent de l'exécuter (« hardware »). Selon cette définition, le premier ordinateur est conçu à l'université de Manchester par l'équipe dans laquelle travaille Alan Turing. Le premier modèle expérimental, la Small-Scale Experimental Machine (ou « baby ») est réalisé en juillet 1948. Il sera amélioré pour donner le Manchester Mark I achevé en avril 1949. Cet ordinateur fut popularisé par la presse britannique¹³ qui inventa à son sujet le terme de « cerveau électronique ». Cette dénomination allait initier une polémique au sujet d'une possible « intelligence artificielle ». En réponse à la diatribe du neurochirurgien Sir Geoffroy Jefferson qui déclarait :

Tant qu'une machine ne pourra pas écrire un sonnet ou composer un concerto en raison des pensées et des émotions qu'elle ressent, et non par le tirage au sort de symboles, nous ne pourrons pas admettre qu'une machine égale un cerveau c'est-à-dire, non seulement l'écrire, mais aussi savoir qu'elle l'a écrit.¹⁴

Turing déplace la question en reportant l'intelligence artificielle, non sur une reconstruction de l'intelligence humaine, mais sur une « imitation » de celle-ci validée, non par les processus techniques mis en jeu, mais par les conséquences des productions de la machine sur un sujet humain. C'est le très célèbre test de Turing¹⁵ dans lequel, par un jeu de questions réponses, un expérimentateur doit décider laquelle des deux entités qu'il ne voit pas est un homme et laquelle est une machine. Le programme du test de Turing ne reproduit pas les processus cognitifs et affectifs qui président à la créativité humaine, ce n'est pas une simulation de l'intelligence comme tentera ensuite de l'être l'intelligence artificielle par la voie du connexionnisme.¹⁶ Le test de Turing constitue plutôt un simulacre d'intelligence, c'est-à-dire qu'il en simule le produit créatif, non le

processus créatif. Il ne s'agit dès lors plus vraiment d'intelligence mais de performance : « Le test de Turing vaut d'abord par le but empirique qu'il assigne à l'IA : faire que la machine rivalise par ses performances avec l'humain dans différents registres réputés requérir de l'intelligence. »¹⁷ N'est-ce pas là le but de tout outil ? Si le marteau était aussi peu performant que ma main pour enfoncer un clou, à quoi me servirait-il ? Et comme pour tout outil, les performances ne sont atteintes que dans des domaines limités et parfaitement définis (essayez de faire faire du vélo à un marteau !)

Le Manchester Mark I a servi de prototype au tout premier ordinateur commercial, le Feranti Mark I dont Christopher Strachey a développé le programme. Strachey connaissait bien Turing et était influencé par ses idées. C'est ainsi qu'il développa en juin 1952, soit 9 ans avant la naissance de l'Oulipo, le premier générateur informatique de textes, *love letter*, conçu comme une application qui permettait de valider la conception de l'intelligence artificielle selon Turing. Strachey ne semble pas avoir accordé une grande importance à ce programme. Il ne le mentionne pour la première fois dans un article qu'en 1954,¹⁸ et la présentation qu'il en fait occupe moins d'une page de l'article. En effet, un programme de jeu de dames qu'il avait réalisé en 1951 était bien plus convaincant. Ce programme de jeu de dames est d'ailleurs le plus long qui ait été créé sur le Manchester Mark I et c'est lui qui avait rendu Strachey célèbre dans le petit milieu naissant des informaticiens.

Dans cet article, Strachey répond à l'objection de Jefferson et à la peur naissante des « machines pensantes » en affirmant dès le début que les machines ne pensent pas, qu'elles ne peuvent faire que ce qui est prévu dans le programme, bêtement, « même si le processus semble fantastiquement compliqué et laborieux ».¹⁹ Il insiste sur le fait que le programme est écrit par un humain et que « tout se passe comme si l'ensemble de la pensée requise pour solutionner le problème est concentré dans le programme », la machine n'en étant qu'un esclave incroyablement rapide mais dépourvu d'intelligence : elle ne fait preuve ni de jugement, ni d'initiative, ni d'esprit critique sur la justesse des résultats obtenus. Il a une très belle image pour illustrer la place du programme : « sans lui, l'ordinateur est comme une machine à écrire sans dactylo ou comme un piano sans pianiste ».²⁰ Strachey était conscient de la dimension de simulacre (il utilise le mot « ruse ») du programme de lettres d'amour :

It is clear that these letters are produced by a rather simple trick and that the computer is not really « thinking » at all. This is true of all programs which make the computer appear to think; on analysis they are nothing more than rather complicated tricks. However, sometimes these tricks can lead to quite unexpected and interesting results.²¹

La question de la machine pensante est posée en France à la même époque par Louis Couffignal et Albert Ducrocq. Couffignal avait rencontré Norbert Wiener dès 1946 et a fortement contribué au développement de la cybernétique en France. Il publie son ouvrage « les machines à penser »²² en 1952 et participe au développement de la machine Calliope²³ d'Albert Ducrocq capable de générer des poèmes. En fait, Calliope n'est pas un programme informatique mais un robot électronique. Cette machine était encore en état de marche à la mort d'Albert Ducrocq en 2001. Tout comme pour Strachey, c'est bien la question de l'intelligence artificielle qui motive cette production ; la littérature est instrumentalisée. Cette présentation engendre en France la même peur qu'aux États-Unis. André Parinaud, rédacteur de la revue *Arts* s'inquiète du « robot-poète » d'Albert Ducrocq. Boris Vian lui répond par une nouvelle²⁴ :

Il nous suffit de lire en un journal du matin que M. Albert Ducrocq a construit un robot-poète pour vous étonner aussitôt. Pourtant, qu'est-ce que ça a d'extraordinaire ? Au siècle dernier il y avait déjà Victor Hugo. Alors ?²⁵

b) Le générateur *Love Letter* de Christopher Strachey

Love letter est un générateur combinatoire de lettres d'amour capable de générer $318 \cdot 10^{12}$ (318 000 milliards) lettres d'amour différentes.²⁶ Ces lettres d'amour n'étaient pas créées sur un écran mais imprimées sur la notice de l'ordinateur. Le programme a fonctionné semble-t-il d'août 1953 à Juillet 1954²⁷ au rythme de production d'une lettre par jour. En voici un exemple tiré de l'article de Christopher Strachey de 1954 :

Darling Sweetheart
You are my avid fellow feeling. My affection curiously clings to

your passionate wish. My liking yearns for your heart. You are my
wistful sympathy: my tender liking.
Yours beautifully
M. U. C.

Ce programme implémente la structure des phrases à trou, l'algorithme qui sera de loin le plus utilisé pour réaliser des générateurs combinatoires durant cette première période de la littérature numérique. Cet algorithme utilise une phrase type dans laquelle on remplace des éléments de vocabulaire par d'autres choisis aléatoirement parmi des listes, une par emplacement considéré. Cet algorithme, très simple, reporte l'accord grammatical sur les listes elles-mêmes (le vocabulaire de ces listes doit être accordé en fonction du contexte syntaxique dans lequel un élément de la liste se trouve dans la phrase) et reporte la gestion de la signification sur le choix du vocabulaire composant chaque liste. Bien évidemment, chaque liste correspond à une fonction syntaxique précise et ne comporte qu'un seul type de mots : substantifs, adjectifs ou verbes. Le programme de Strachey utilisait 2 types de phrases qu'il choisissait de façon aléatoire de façon à composer une lettre de 5 phrases. Cette structure était précédée d'un en-tête lui-même construit aléatoirement et se terminait par la signature M.U.C, acronyme de Manchester University Computer. Voici comment Strachey décrit cette structure :

Apart from the beginning and the ending of the letters, there are only two basic types of sentence. The first is « *My* ô (adj.) ô (noun) ô (adv.) ô (verb) *your* ô (adj.) ô (noun). » There are lists of appropriate adjectives, nouns, adverbs, and verbs from which the blanks are filled in at random. There is also a further random choice as to whether or not the adjectives and adverb are included at all. The second type is simply « *You are my* ô (adj.) ô (noun), » and in this case the adjective is always present. There is a random choice of which type of sentence is to be used, but if there are two consecutive sentences of the second type, the first ends with a colon (unfortunately the teleprinter of the computer had no comma) and the initial « *You are* » of the second is omitted. The letter starts with two words chosen from the special lists; there are then five sentences of one of the two basic types, and the letter ends « *Yours* ô (adv.) M. U. C. »²⁸

Strachey ne semble pas non plus accorder d'importance aux particularités textuelles ou littéraires des lettres d'amour générées. Il admet avoir utilisé un thesaurus de langue anglaise pour le choix du vocabulaire. Celui-ci ne semble donc pas correspondre à une motivation particulière. Il admet également le caractère rudimentaire de la structure. Le principal intérêt du générateur, selon lui, réside dans l'écart entre le peu de sophistication de l'algorithme et la grande diversité des lettres produites, preuve justement du simulacre. C'est sans doute la raison pour laquelle ce générateur, bien que présenté dans une revue culturelle réputée, n'est généralement pas considéré comme la première œuvre de littérature numérique : les textes générés n'ont pas été publiés, sa présentation publique intervient 2 ans après sa réalisation et elle est surtout centrée sur les aspects culturels (la question de l'intelligence artificielle) alors que la motivation littéraire en semble absente. Toutefois ce générateur est repéré outre atlantique comme une pièce importante dans l'étude de genre : Strachey et Turing étant tous deux homosexuels, générer des lettres d'amour au vu et au su de tous, dans lesquelles ni le genre ni le mot amour n'apparaissent, n'a rien d'anodin. J.R. Carpenter, dans sa thèse de doctorat en cours, examine les documents laissés par Strachey et tente de démontrer l'existence d'une littéarité dans ces lettres, notamment par leur dimension de pastiches et par le fait qu'elles ont connu, semble-t-il, une forme de publication publique sur des panneaux d'affichage au sein de l'université. Il se pourrait donc qu'à l'issue de ce travail, la place de cette production dans l'histoire de la littérature numérique soit réévaluée. En tout état de cause, on ne peut nier qu'il s'agisse bien là de la première rencontre entre informatique et textualité.

Remarquons également que cette première production se fait en dehors de toute référence à la question littéraire de la combinatoire de texte, en dehors d'ailleurs de toute réflexion sur les questions littéraires. Aucun enjeu littéraire ou artistique n'est exprimé dans le contexte de cette production. C'est la raison principale à mon sens pour laquelle elle n'est pas considérée comme la première œuvre littéraire numérique.

c) Le robot poète d'Albert Ducrocq

Le principe de fonctionnement de Calliope est relativement éloigné de celui de la phrase à trou et s'approche par certains aspects de l'algorithmique du générateur automatique de texte qui sera mis au point par Jean-Pierre Balpe dans

les années 1980.

Calliope ne produit pas des phrases lisibles par un humain mais un énoncé en langage binaire. Le robot produit en effet une série aléatoire de flashes rouges (représentant les chiffres 1) et verts (représentant les chiffres 0) au rythme d'un chiffre par seconde. Les messages ainsi formés doivent être interprétés par un 'lecteur' humain en fonction d'un code précis arbitraire dans lequel le vocabulaire est codé selon une arborescence sémique, chaque niveau de l'arbre étant écrit en reprenant le code du niveau supérieur. Ainsi, par exemple, la séquence composée du seul chiffre 1 désigne un être. Le code « 11 » désigne un être animé, le code « 10 » un être inanimé. Ainsi, le mot « animal » sera codé par la séquence « 1110 », le mammifère par « 1110 11 », le mot « tigre » par « 1110 11 1100 10 »

Voici quelques règles définissant ce code :

ô les qualificatifs comme les adjectifs sont directement associés aux substantifs et les articles ne sont pas pris en compte. En clair, le langage code des groupes nominaux. Ducrocq note « en langage logique, la notion n'a plus son sens grammatical habituel puisque les qualificatifs sont incorporés aux idées. » Les blocs ainsi formés constituent les « êtres », ils débutent tous par le chiffre 1 ;

ô les verbes et adverbes sont groupés sous le terme d'instructions ; elles débutent toujours par le chiffre 0 ;

ô tout être ou instruction débute par un radical de 4 chiffres qui indique sa classe fondamentale ;

ô les phrases se présentent sous la forme d'un groupe « être ó instruction ó être » ;

ô les phrases ou propositions sont séparées par la séquence 0000 et, au sein d'une phrase, l'instruction est amorcée par une succession de trois 0 ou, s'il n'y en a pas, de deux 0 (si ces suites n'existent pas, la phrase est lue comme dépourvue de verbe).

Avec ces règles arbitraires mais qui procèdent d'une certaine logique, la longueur minimale d'un mot est de 4 chiffres. On ne peut donc pas attribuer de signification à une séquence plus courte. Ainsi, la séquence 100010 ne peut pas s'interpréter comme une séquence être-instruction malgré la présence successive de trois 0, car le premier terme, censé représenter l'être, ne possède qu'un seul chiffre et non un radical. La séquence ne peut donc s'interpréter que comme un terme de la catégorie « être ». Ducrocq remarque que la longueur maximale d'un

mot est de 16 chiffres, valeur suffisante pour coder 65 000 mots, soit plus de mots que les dictionnaires usuels. Il évalue à 24 chiffres la longueur moyenne d'une entité et à 16 celle d'une instruction, de sorte que la longueur moyenne d'une phrase énoncée par Calliope est de 65 chiffres. Remarquons que comme ni le texte, ni les « mots » du langage binaire ne comportent pas marqueur de fin, la signification du dernier terme dépend de la date à laquelle on arrête la machine. Par exemple, si le message se termine par 1110 et que le contexte permet d'interpréter cette fin comme étant le mot « animal », il aurait pu se faire qu'en arrêtant la machine deux secondes plus tard elle ait produit la suite 11, et dans ce cas la séquence finale aurait été 111011 qui se traduit par « mammifère ». Évidemment, comme les termes sont écrits en décomposition sémique, on gagne en précision sans modifier fondamentalement le concept évoqué, mais, d'un point de vue poétique, cela peut produire une différence importante.

Ducrocq développe en détail l'interprétation de la séquence suivante produite par Calliope dans laquelle j'ai porté en gras souligné la séquence qui sépare les propositions et en souligné les séquences qui débudent les instructions : 11000111011100001000100100101000011101000110110110010001001111.

Voici ce qu'il en dit :

La première [phrase] est extrêmement courte puisqu'elle se réduit à un mot, et le dictionnaire binaire nous donnera tout de suite la traduction = appréciation. Considérons maintenant la phrase principale de ce texte. Il s'agit de prêter attention aux blocs de trois 0 en ne tenant pas compte de celui suivant le 1, signe initial de la phrase car il fait partie du radical (1000) par lequel commence le premier mot de cette phrase. [í] Pour le sujet, nous trouvons en effet que le mot 10001001001 veut dire tissu, tandis que le court qualificatif 01 est un terme très vague voulant seulement indiquer l'idée abstraite et l'idée de mouvement. On pourra alors dire quelque chose comme « un tissu mouvant ». Si nous passons ensuite au verbe, on devra supprimer le premier 0 qui est seulement le signal indiquant la transition entre l'être et l'instruction. Par ailleurs, le radical 0011 concerne la rubrique « logique » et dans celle-ci les signes 101 expriment l'idée de commander. Finalement l'ensemble du texte proposé sera traduit par une phrase telle que « Appréciation : un tissu mouvant

commande, se soumet et emprisonné ». C'est évidemment du pur surréalisme.²⁹

On voit sur cet exemple l'énorme différence entre les énoncés de Calliope et les textes générés par ordinateur : ces derniers sont réellement des énoncés en langue naturelle alors que les « poèmes » créés par Calliope sont en fait des verbalisations d'énoncés écrits en « langue binaire ». Ils résultent donc d'une traduction humaine, d'autant que ni les articles, ni la ponctuation ne sont codés en langage binaire. Le début de l'exemple précédent aurait donc tout aussi bien pu être traduit par le poème suivant en fonction du décodage décrit par Ducrocq « Appréciation ; tissu d'abstraite mouvance tu commandesí » qui me semble moins plat que la solution proposée par Ducrocq. Autrement dit, contrairement à un texte généré, la qualité textuelle, au regard de règles d'appréciation d'un texte imprimé du genre poème surréaliste, dépend plus du « traducteur » que de calliope. On est finalement assez loin du robot poète

Voici quelques exemples de poésie « écrite » par Calliope :

La victoire sera ardue. Une mince couche de neige très étendue a couvert le champ de blé blanc rendu vigoureux : la mine sacrée gît en dessous. La côte monte à l'assaut de la vérité ; ses dessins variés forment une chaîne qui brille malgré son poison, tandis que l'aride divagation des mois et des ans a donné au buis la vigueur du vieil arbre.³⁰

d) Le test de Couffignal

Louis Couffignal réalise avec un ingénieur une machine programmée qui semble reprendre le principe d'écriture mot à mot des phrases mais sous forme informatique, ce qui le rend plus proche des générateurs automatiques que la machine de Ducrocq. Il reprend d'ailleurs le nom de Calliope. Voici comment Couffignal décrit le fonctionnement de sa machine :

On avait mis dans la machine un certain vocabulaire et les règles de la grammaire, de façon qu'elle fasse des phrases correctes. Mais le choix des mots et, si l'on admet cette image, le choix des idées que ces mots

exprimeront, était entièrement laissé au choix de la machine. [í] On a mis dans la mémoire de la machine un lexique composé de trois cents mots de la langue usuelle et d'une certaine teneur littéraire. On a mis également dans sa mémoire les règles de grammaire et les règles de la syntaxe grammaticale, de sorte qu'elle ne puisse choisir, pour faire suite à un mot dans une phrase, qu'un mot qui s'accorde grammaticalement avec elle et qui donne une phrase ayant un sens et étant correct grammaticalement. Puis, on a également donné à la machine une suite de nombres ou de chiffres, qui sont choisis absolument au hasard et qui se suivent au hasard. La machine a choisi au hasard, sous la réserve de la correction grammaticale dont j'ai parlé, dans son lexique un mot qui pouvait faire suite correctement, au point de vue grammatical, au mot précédent, ce qui devint une phrase. Lorsqu'elle a choisi un point ou une virgule, elle a mis un point comme signe lexical pour terminer une phrase et en recommencer une autre. Par conséquent, la base du travail soi-disant intellectuel de la machine est donc un choix au hasard parmi les mots d'un lexique, sous la réserve que ce mot puisse suivre le précédent dans les conditions de la syntaxe grammaticale de la langue courante.³¹

Il met au point un test qui sera répété au moins 3 fois et qui est même expérimenté aux rencontres internationales de Genève en 1965, générant un débat³² qui éclaire magistralement la conception qui prévaut à l'époque. Ce test consiste à présenter au public 2 poèmes : l'un fabriqué par Calliope, l'autre étant la 3^e strophe d'un long poème d'Eluard³³ et à poser deux questions³⁴ :

- ô lequel des deux textes jugez-vous le plus poétique ?
- ô lequel des deux textes vous paraît écrit par la machine ?

Le poème utilisé pour ce test est le plus connu de ceux composés par la machine de Couffignal. Il a vraisemblablement été créé en 1964.³⁵ Le voici tel qu'il apparaît dans les actes des rencontres internationales :

Un doute agréable couleur de lotus endormi entretient
la joie sur cette île montagneuse.

Il instruit avec un retard utile et propose plusieurs
voies pour aboutir à la solution attendue.

L'éternité dure une heure.

Les pieds multiples d'un char qui se dresse comme une forteresse seront précieux demain pour convaincre qu'il faut labourer dans le détail l'île où pousse le paisible tilleul.

Ainsi la vie est fluide, la grêle de mai arrache le lierre pour commander un nouveau décor.

Un rideau de plantes rouges meuble l'éternité.

Le hérisson avance péniblement ; le corail rêve ; la vipère pourrie sur laquelle est tombé le marteau étale ses organes sous le genévrier tandis que le blé germe ; le morse essoufflé arrive devant la forteresse ; et l'amphioxus voit le soleil chaque jour du mois. Tous aiment peindre la terre.

Plus tard, sur une écorce, le ver luisant nous conseillera pour vaincre la chair.

Couffignal précise :

J'ai choisi ce poème parce qu'il me donnait un texte suffisamment voisin de l'autre pour que la difficulté de choix puisse se présenter, mais je n'ai aucunement l'idée de comparer les poèmes d'Eluard à ceux faits par la machine.

Il précise également dans quelle condition la machine a généré ce texte :

La machine a écrit des centaines de poèmes dans les conditions suivantes : on l'a simplement mise en route pendant une semaine environ. Un huissier a contrôlé et constaté que la machine avait travaillé seule, sans aide humaine. Elle a donné environ cent cinquante textes. Pour la longueur de ces textes, quand elle paraissait suffisante, on coupait tout simplement, et le lendemain, ou quelques heures après, la machine recommençait. Sur ces

cent cinquante textes, c'est moi qui ai choisi celui-ci. J'avais sous la main le recueil d'Éluard et j'ai cherché, pour faire mon expérience, deux textes suffisamment voisins pour qu'il puisse y avoir une difficulté à faire le choix. L'expérience a été montée dans ces conditions. J'ai trouvé que les deux textes présentaient suffisamment de parenté pour que le jugement puisse se répartir, et c'est effectivement ce qui s'est passé dans les trois expériences qui ont été faites.³⁶

En général, la moitié environ des sondés sur les 3 expériences estime que le texte de Calliope est le plus poétique, sans savoir qu'il s'agit du texte créé par la machine. Le résultat de la seconde question du test n'est donné que pour l'expérience réalisée lors des rencontres : cinquante personnes sur quatre-vingt ont correctement trouvé lequel des deux poèmes avait été écrit par la machine. Il est intéressant de noter que parmi ceux-ci, certains ont pourtant considéré que le poème écrit par Calliope était plus poétique que le poème d'Éluard. L'assistance en a conclu que le caractère « poétique » de l'œuvre était de l'ordre de la réception et non le fait de l'auteur. Il est également remarquable que Couffignal n'accorde aucune importance à la taille du vocabulaire, pas plus qu'à sa nature. À la remarque du président de séance qui précise que le poème généré comporte cent dix mots, soit approximativement la moitié du vocabulaire connu de la machine, et que donc le choix de ce vocabulaire par le programmeur est une donnée importante de la poéticité du texte généré, Couffignal répond :

Ceci est très vrai, mais je ne crois pas que cela ait une grande importance en ce qui concerne la valeur de l'expérience, car on pourrait avec des machines plus modernes [] faire une liste de mots qui comprenne tout le vocabulaire. On pourrait également les choisir [] Si on avait pris, à la place, une langue de spécialité pour la construction, la machine aurait fait un projet de machine au lieu de faire des poèmes. Le vocabulaire mis dans la machine conditionne le produit [] Cette expérience montre qu'une machine peut aussi faire de la production artistique.³⁷

Il prédit même au passage ce qui sera une grande réussite des premiers générateurs automatiques de texte : « si on mettait dans la machine une règle plus complexe, une règle de stylistique que plus ou moins consciemment suit un auteur, la machine ferait des textes à la manière de l'auteur choisi. »³⁸

4. Un basculement de la problématique

a) *Stochastique* texte de Theo Lutz

En 1959, Théo Lutz, un élève de Max Bense, publie dans la revue *Augenblick* le programme d'un générateur combinatoire de textes d'une ligne ainsi qu'une sélection de textes générés sous le titre *Stochastische Texte*.³⁹ Le générateur avait été programmé sur un ZUSE Z 22 au centre informatique de Stuttgart. Comme pour le générateur de Strachey, les textes générés étaient imprimés par un teleprinter. Lutz a calculé que son programme pouvait générer 4174304 textes, chacun n'étant constitué que d'une phrase. Il n'en a produit en réalité que 50 dont 35 seulement furent imprimés dans l'article. L'algorithme de cette œuvre est voisin de celui utilisé par Christopher Strachey et l'article de Lutz est purement technique.⁴⁰ Il réaffirme également le potentiel artistique de la machine et sa capacité à reproduire le style d'un auteur en allant plus loin que Couffignal car il donne un algorithme de statistique textuelle pour décrire le style d'un auteur à partir de la fréquence des mots. Il ne va toutefois pas jusqu'à implémenter cette extension, son générateur reste des plus basiques.

Le générateur de Lutz est donc très proche, dans sa présentation, de ceux qui l'ont précédé. Et pourtant cette publication est souvent considérée comme l'acte de naissance de la poésie numérique et, par-là même, de la littérature numérique. Bien que Théo Lutz ne semble pas en avoir eu conscience, on peut en effet la considérer, non comme l'illustration de l'intelligence artificielle, mais comme une démonstration de la conception de Max Bense selon laquelle une œuvre se définit par l'association de son original et de l'ensemble de ses variantes. Cette démonstration est d'autant meilleure qu'il n'y a pas de texte généré « original ». C'est sans doute selon cette conception de Max Bense qu'il faut comprendre l'intérêt que porte l'article au calcul précis du nombre de variantes. Par ailleurs, le vocabulaire utilisé est extrait du *château* de Kafka (16 substantifs et 16 adjectifs dont Théo Lutz donne la liste dans l'article) et *Augenblick* est la revue de Max Bense, une des plus importantes revues d'avant-garde de l'époque. Le contexte dans lequel s'inscrit le générateur de Théo Lutz recentre ainsi ce qui sera ultérieurement dénommée « littérature numérique » sur la question du texte. La question de l'auteur n'est pas abordée dans l'article. Voici quelques textes générés publiés dans l'article de Lutz :

NICHT JEDER BLICK IST NAH. KEIN DORF IST SPAET.
 EIN SCHLOSS IM FREI UND JEDER BAUER IST FERN.
 JEDER FREMDE IST FERN. EIN TAG IST SPAET.

b) Les questions littéraires posées par l'approche scientifique

Ainsi donc, il semble bien que la rencontre entre le numérique et la littérature se soit faite indépendamment aux États-unis et en France autour de la question de l'intelligence artificielle. Elle pose fondamentalement la question de l'auteur : a-t-on affaire à un « robot-poète » ? Simulacre ou non, considérer le texte généré comme un texte imprimé revient à attribuer à la machine la qualité d'auteur. Cette équation : « la machine est l'auteur » aura vite fait de faire oublier l'affirmation de Strachey, exacte, selon laquelle la machine n'est rien sans le programmeur. C'est cette équivalence supposée entre la machine et l'auteur qui rendra la génération de texte si scandaleuse aux yeux de beaucoup, bien qu'à ma connaissance aucun auteur ne l'ait jamais revendiquée.

Le débat de 1965 autour de Calliope n'aborde pas directement la question de l'auteur mais celle du rôle de l'humain dans la génération. Les protagonistes en arrivent rapidement à la conclusion que l'humain joue un rôle fondamental dans le choix du vocabulaire et des règles implémentées dans l'algorithme :

Certainement, avant tout ceci est intervenu celui qui a fait la machine, et beaucoup d'autres personnes : les grammairiens qui ont fait les règles de grammaire et la société humaine qui a fait le langage poétique. C'est déjà un instrument humain que vous mettez dans la machine, car lorsque vous mettez dans la machine « coucher de soleil », c'est déjà un mot qu'il suffit de prononcer pour vous émouvoir.⁴¹

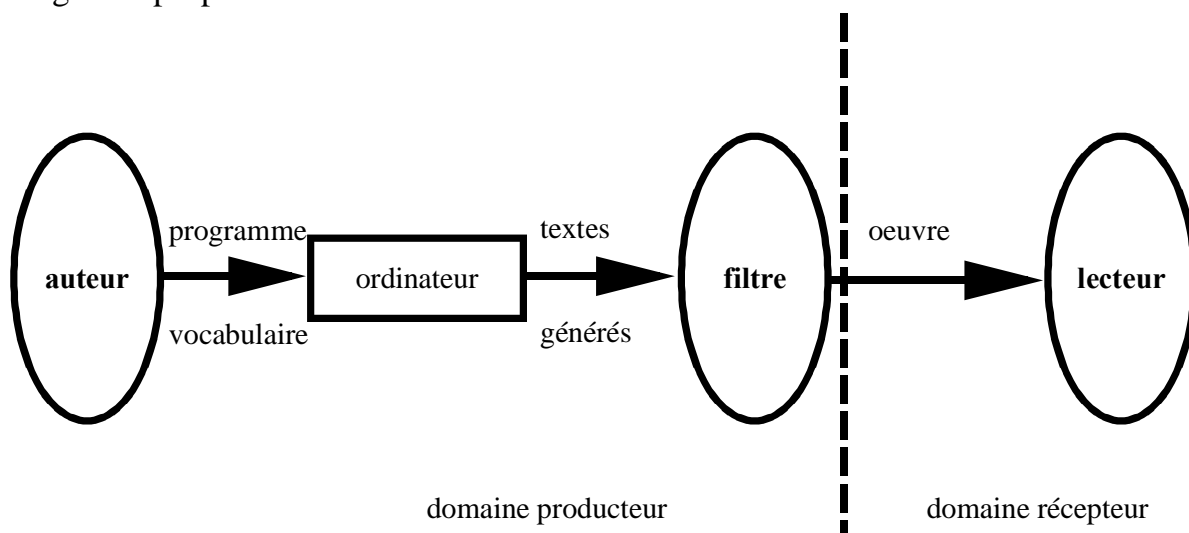
Et c'est le poète Pericle Patochi qui énonce le mieux la dimension humaine implémentée dans la machine :

Je pense que cette machine qui a reçu un bagage de paroles, a reçu quelque chose qui vient de ce langage vivant qu'est le langage des vivants, qui est une grande âme parlante. Toutes les paroles, toutes les langues, à travers les voyages sémantiques, sont l'expression de ce que nous sommes. Or, que cette machine ait précisément réussi à nous toucher, à nous émouvoir, cela est dû au fait que la machine a reçu ce souffle et qu'elle nous le rend.⁴²

La question de l'auteur, question autant socio-culturelle que littéraire, va jouer un rôle important dans cette période. C'est peut-être elle qui, au fond, a incité des auteurs à s'engouffrer dans la génération combinatoire, et non la question textuelle en elle-même, contrairement à l'impression qu'on en a aujourd'hui, depuis que l'Oulipo est devenu le symbole paradigmatique de cette période. Le questionnement de la notion d'auteur par la génération de texte va d'ailleurs se révéler dans une affaire judiciaire⁴³ : Jean Baudot publie en 1964 à Montréal, l'un des plus célèbres recueils de poèmes générés par ordinateur : *La machine à écrire*.⁴⁴ Pierre Moretti, un scénographe de théâtre, demande en 1967 à Jean Baudot d'utiliser son générateur pour fabriquer des répliques pour sa pièce *équation pour un homme actuel*.⁴⁵ Pierre Moretti sélectionne le texte dans un échantillon de plusieurs milliers de phrases⁴⁶ composées à partir d'un vocabulaire d'environ 8000 mots.⁴⁷ La pièce est censurée pour atteinte aux bonnes mœurs et toute la troupe des Saltimbanques se retrouve devant le juge. La question est bien sûr posée : qui est l'auteur de ces répliques sulfureuses ? Un avocat de la défense a alors suggéré de mettre « l'ordinatrice » en prison. Il n'a semble-t-il pas été suivi puisque les comédiens ont passé la nuit au poste à cause de leur jeu scénique.

Toutes les œuvres textuelles produites avec un ordinateur, jusqu'au milieu des années 1980, sont des générateurs de textes. Chris Funkhouser⁴⁸ et Ambroise Barras⁴⁹ mentionnent un certain nombre de générateurs de textes réalisés dans les années 1960-1970. Le dispositif de l'écran n'étant pas encore d'usage courant, les textes générés sont imprimés. L'anthologie Bailey⁵⁰ est l'édition la plus connue et la plus complète de cette production. Ainsi, le dispositif de l'imprimerie demeurant le dispositif de diffusion, la question de la lecture n'est pas posée, pas plus que celle de l'œuvre : dans les conceptions de l'époque, le seul texte généré imprimé et proposé à la lecture constitue l'œuvre. Or les textes générés proposés à la lecture sont loin de refléter l'ensemble des caractéristiques de la génération et ce pour deux raisons. La principale est que l'auteur joue lui-même un rôle de filtre entre ce que produit la machine et ce qui est fourni au lecteur : le texte généré imprimé est choisi, voire restructuré avant impression de façon à extraire de la masse produite les « heureux hasards » qui, seuls, accèdent au statut d'œuvre. Par ailleurs, seul un petit nombre de textes est sélectionné et en aucun cas le générateur ne produit la totalité des textes que son programme est susceptible de produire.

La situation de communication entre l'auteur du programme et le lecteur du texte final est donc globalement la suivante, qui confine la machine au rang d'outil de production et permet de qualifier la production de cette période de « littérature assistée par ordinateur ». Dans ce schéma nous avons mentionné les rôles « auteur », « filtre », « lecteur » tenus par les sujets humains. Remarquons l'absence de feed-back entre les rôles filtre et auteur : le filtre établit un choix a posteriori dans la production mécanique mais ne modifie pas le programme pour améliorer cette production. N'oublions pas qu'à cette époque l'accès aux ordinateurs n'était pas commode et que le programmeur n'était pas toujours le concepteur de l'œuvre, ce dernier étant en revanche toujours le filtre signant le texte généré proposé à la lecture.



[FIGURE 1]

Schéma de communication de la littérature assistée par ordinateur

©Philippe Bootz, tous droits réservés.

c) Le rôle de l'Oulipo et la question textuelle

L'Oulipo⁵¹ est créé en 1961, il n'est donc pas un précurseur de la génération informatique comme la plupart des gens le pensent souvent. L'Oulipo, on le sait, donne ses lettres de noblesse à la combinatoire, d'une part en la replaçant dans la perspective de l'histoire de la littérature et en montrant que la combinatoire remonte au XV^e, d'autre part en en faisant une marque stylistique. Il est pourtant

étonnant et quelque peu paradoxal que l'Oulipo devienne le mouvement paradigmatique de la génération informatique de texte, et ce pour plusieurs raisons. La première est que l'Oulipo ne s'intéresse que très tardivement à l'informatique : ce n'est qu'en 1975 que Paul Braffort réalise une version programmée des *cent mille milliards de poèmes* de Queneau pour l'exposition Europalia. La relation entre l'Oulipo et l'informatique ne consiste pas à utiliser l'ordinateur pour créer de nouvelles contraintes, mais à créer des versions numériques d'œuvres à contrainte existantes par ailleurs : il s'agit d'un travail de remédiatisation, non de création originale. Cela va favoriser l'idée qu'au fond il n'y a pas de différence fondamentale entre un texte généré selon un procédé analogique (en clair en utilisant du papier et un crayon) et un texte généré par ordinateur. On en conclut alors que la littérature numérique, nécessairement combinatoire à l'époque, est potentielle et que l'ordinateur n'est qu'un outil qui permet d'explorer de façon exhaustive les textes potentiels. Elle perd grandement de son intérêt car ce qui caractérise le potentiel c'est sa dimension de « possible » au sens de Deleuze⁵² et Pierre Lévy :

Le possible est déjà constitué mais il se tient dans les limbes. Le possible se réalisera sans que rien ne change dans sa détermination ni dans sa nature. C'est un réel fantomatique, latent. Le possible est exactement comme le réel : il ne lui manque que l'existence.⁵³

En focalisant sur la structure plutôt que sur le texte, c'est-à-dire en considérant que tous les textes qu'une structure peut générer sont équivalents, l'Oulipo induit, sans doute à son corps défendant, une vision culturelle du texte : le texte (au sens d'entreprise créative et non au sens d'un énoncé) se situe dans la structure. Autrement dit, transposé en termes informatiques : le texte = l'algorithme. Or les générateurs combinatoires informatiques, y compris ceux réalisés par des membres de l'Oulipo, ne se sont pas attachés à travailler sur des structures nouvelles ou complexes, tout au plus ont-ils transposé des travaux antérieurs ou créé des haïkus ou des poèmes permutationnels. Le plus souvent les textes générés avaient un caractère plus ou moins surréaliste parce que le niveau sémantique n'y était pas géré, caractéristique qu'ils partagent avec les autres recherches oulipiennes. Les générateurs combinatoires se sont surtout penchés sur la programmation algorithmique des deux dimensions grammaticales que sont l'axe syntagmatique et l'axe paradigmatique. Et si la

génération combinatoire informatique s'accorde bien en effet avec la conception de Max Bense, qui noie effectivement le « génie » humain créateur d'unicité dans la masse de la quantité qui caractérise l'art de consommation, elle n'a pas servi à une exploration de structures plus poussée que celle entreprise par d'autres moyens.

En revanche, l'Oulipo aura le mérite de faire prendre conscience, en France, que la relation entre littérature et informatique existe bel et bien et qu'elle est suffisamment complexe pour qu'on s'y attache de façon spécifique. C'est de cette constatation que naîtra l'A.L.A.M.O. Il aura également le mérite de créer, au niveau international, un véritable courant artistique fondé sur la mise en exergue de la structure.

d) L'A.L.A.M.O. : une idée tardive

L'A.L.A.M.O.⁵⁴ (Atelier de Littérature Assistée par la Mathématique et les Ordinateurs) voit le jour en 1982. Il est créé par des membres de l'Oulipo auxquels s'ajoutent d'autres personnes, dont Jean-Pierre Balpe qui le quittera en 1985. L'A.L.A.M.O. pense à ouvrir pour une littérature numérique, mais encore considérée comme « à venir ». Jean-Pierre Balpe écrit ainsi en 1984 dans un numéro d'*Action poétique* consacré à l'A.L.A.M.O. :

Ce numéro de notre revue ne veut donc pas présenter un bilan, le résultat abouti des réalisations d'un groupe, mais plutôt marquer le début de tentatives nouvelles d'écriture, jeter les bases de ce qui pourra, à côté de modes d'écriture plus 'conventionnels' être une des voies où s'engagera certainement la littérature.⁵⁵

Pourtant, l'A.L.A.M.O. n'ira pas plus loin que l'Oulipo dans sa mise en relation entre informatique et littérature. Il reste attaché à la conception d'une littérature « assistée » et ne s'intéresse qu'à la gestion syntaxique de la langue. Le meilleur exemple de production de l'A.L.A.M.O. est peut-être le projet LAPAL⁵⁶ créé à l'initiative de Paul Braffort. Ce projet se veut un outil de création de textes indépendant des technologies utilisées.

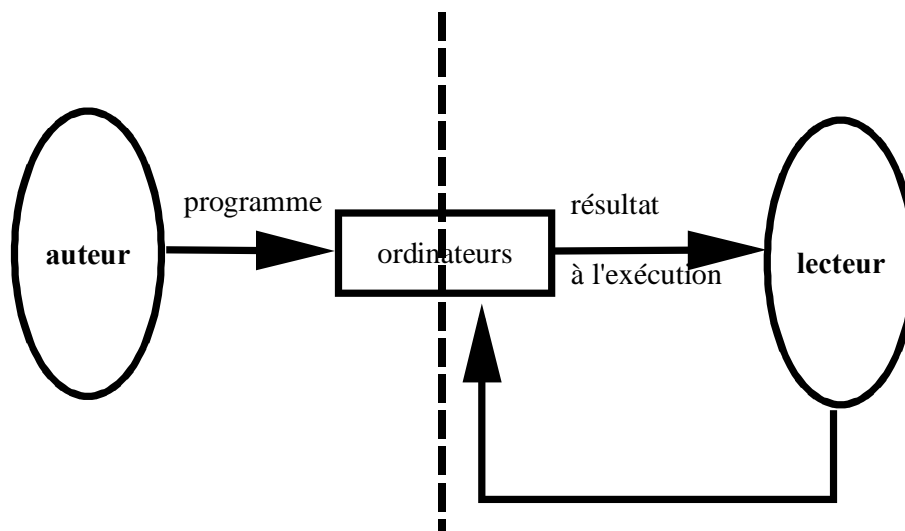
5. Les littératures pour l'écran informatique

a) Le dispositif écranique

L'A.L.A.M.O. ne perçoit pas que la nature même du dispositif technologique informatique est en pleine mutation et que la situation de communication sur laquelle repose cette conception devient caduque, notamment par le fait que le dispositif de diffusion des productions numériques n'est plus l'imprimé mais l'écran. La littérature sur écran devient possible avec l'apparition, au milieu des années 1970 de l'informatique familiale. Il faut en effet que plusieurs conditions soient remplies pour qu'une littérature numérique de l'écran puisse émerger, au moins selon des circuits de distribution et non en performance dans des galeries. Ces conditions sont : un parc d'ordinateurs chez les particuliers, ce qui suppose un prix raisonnable pour le matériel ; une qualité graphique (résolution, nombre de couleurs) suffisante, et l'existence de logiciels de création, ou de langages de programmation, qui permettent à des auteurs de pratiquer par eux-mêmes la création numérique sans être pour autant des professionnels de l'informatique. Ces conditions sont réalisées avec la micro-informatique familiale, dont la machine emblématique est l'Apple II créé en 1977. La dernière condition à remplir est d'avoir un parc de machines compatibles. Cette condition sera remplie avec la création du PC en 1981. Au milieu des années 1980, deux standards ont réussi à s'imposer et à effacer les autres : le Macintosh et le PC. La prise en compte culturelle de ces évolutions se fait, comme toujours, avec un petit temps de retard, et il faut attendre l'apparition des premières entreprises éditoriales pérennes pour que se développe totalement la littérature numérique écranique : 1987 avec la naissance d'Eastgate Systems aux États-Unis qui permettra la création et la diffusion des hypertextes de fiction, la revue *alire* en 1989 en France qui marquera la diffusion de la poésie numérique et permettra une rencontre en ce domaine entre les démarches qui se développent des deux côtés de l'atlantique.

La littérature numérique pour écran modifie profondément le dispositif de communication par l'œuvre en supprimant la fonction de filtre qui restait dévolue à un humain, le programme de l'œuvre s'exécutant en temps réel lors de la lecture. Dès lors, le statut de la machine redevient problématique et la question de l'œuvre doit être posée : quelles sont les frontières de l'œuvre littéraire ? Puisque désormais le programme et le résultat qu'il produit à

l'exécution se trouvent dans le même espace technologique, quel est le rôle de cet espace ? Doit-on « lire » le programme ? Le processus d'exécution joue-t-il un rôle particulier ou n'est-il que l'actualisation de la logique du programme écrit par l'auteur ? L'écran n'est pas un support pérenne : ce qui s'y affiche est transitoire et s'inscrit donc fondamentalement dans la temporalité, alors que ce qui est imprimé s'inscrit fondamentalement dans la spatialité de la page : il y a donc inversion de prééminence entre le temps et l'espace, y compris en littérature. Quelles sont les implications de ce renversement ? Enfin, le lecteur a maintenant, avec l'interactivité, lui aussi la possibilité d'influer l'exécution du programme ; la frontière entre les deux domaines passe maintenant au milieu du dispositif technique : elle se situe entre la machine de l'auteur et celle du lecteur, entre le programme écrit (le programme source des informaticiens) et le programme (compilé ou interprété) exécuté chez le lecteur. Qu'est-ce que cela implique ?



[FIGURE 2]

Schéma de communication de la littérature sur ordinateur « privé »,⁵⁷

© Philippe Bootz, tous droits réservés

Ces problématiques ne sont pourtant pas perçues au niveau culturel avant la création des premières maisons d'édition de littérature numérique à la fin des années 1980. Il y a, comme souvent, un retard entre l'évolution des pratiques imposées par les avancées technologiques et leur imprégnation culturelle.

Un exemple frappant de ce décalage en est donné lors de la grande exposition

« Les Immatériaux » qui s'est déroulée au Centre Pompidou en 1985. Voici ce qu'en dit Jean-Pierre Balpe :

Pour l'anecdote : j'avais réalisé le programme RENGA pour les Immatériaux. Il a produit 32500 et quelques rengas qui ont tous été archivés mais ils n'ont gardé ni le générateur, ni le logiciel de lecture. Ils ont donc quelque part une pile de disquettes que personne ne peut lire. C'est complètement aberrant, mais dans l'esprit de l'époque il y avait cette idée d'òuvre en tant que texte fini ; qu'on garde donc, alors que le générateur de textes est sans importance.⁵⁸

La génération de texte ne répondra pas à l'ensemble des questions précédentes, mais le nouveau dispositif instauré par l'écran en fera évoluer les conceptions.

b) La génération automatique de texte

Jean-Pierre Balpe débute la recherche sur la génération automatique à la fin des années 1970. Ses premiers textes générés sont classiquement imprimés, notamment dans sa revue *Hôtel Continental* comme ce poème réalisé avec son générateur Amour :

Comme je t~~o~~ aime je t~~o~~ aime comme
Comme en enfer je t~~o~~ aime je t~~o~~ aime toujours plongé.⁵⁹

La génération automatique de texte se présente comme une entreprise de transition entre les deux périodes : elle garde des correspondances fortes avec la période précédente, mais elle intègre également des caractéristiques fondamentales de la seconde période. La génération automatique ne s'inscrit pas dans la conception de Max Bense. Il ne s'agit pas, ici, de créer des variantes sur un texte, même si cette étape s'est avérée nécessaire, notamment dans les premiers générateurs, mais de produire des textes en quantité quasi-illimitée sur un sujet donné, possédant une forte cohérence interne sémantique et corrects grammaticalement. Autrement dit, la génération automatique reprend le projet cybernétique, mais en le complexifiant et, surtout, en le rendant opérationnel sur des romans. Jean-Pierre Balpe définit un générateur automatique comme « un

automate capable de produire en quantité psychologiquement illimitée des objets acceptables dans un domaine de communication antérieurement défini, c'est-à-dire reconnu comme domaine par une communauté de récepteurs ».⁶⁰

Le texte généré n'est donc pas une variante mais bien un simulacre. La génération automatique marque toute sa puissance dans la génération de pages de romans, type de texte généré que la génération combinatoire s'avère incapable de réaliser.

La technique du générateur automatique est très différente de celle des générateurs combinatoires : elle ne part pas, tout du moins dans sa version la plus récente, de phrases préexistantes mais d'une description d'un vocabulaire plus ou moins étendu réalisée à l'aide d'attributs dénommés des descripteurs qui gèrent conjointement les dimensions syntaxiques, sémantiques et pragmatiques. Des descripteurs rhétoriques et stylistiques peuvent s'y ajouter. Le programme consiste alors à générer le texte de surface à partir d'une description abstraite, par un moteur d'inférence qui choisit et développe les éléments du texte à l'aide d'une structure hiérarchisée de règles de sélections portant sur ces descripteurs et construisant peu à peu le texte de surface. Le générateur part donc d'un micro-univers et d'un ensemble de relations possibles (notamment syntaxiques et sémantiques).

Balpe développe très rapidement ses générateurs automatiques dans la perspective d'une lecture écran, seule capable, justement, de rendre compte de la quantité quasi-infinie de textes pouvant être générés. La quantité ne vise pas ici à l'exploration ni à l'épuisement d'un domaine, elle signe le refus d'un singulier qui serait figé, elle procède d'une conception dynamique de la littérature, même si elle continue à considérer que le texte généré qui s'affiche de façon statique mais temporaire à l'écran constitue l'œuvre et que le programme source se trouve du côté de l'auteur, restant ainsi implicitement dans la conception d'une littérature assistée.

Ce que travaille la génération automatique, comme les autres formes développées dans cette période, est la lecture et l'écriture. La génération automatique, la première, transpose la littéarité de l'objet texte sur les fonctions qui manipulent cet objet : le texte est peut-être un simulacre, mais ce simulacre, parce qu'il est mouvant, imprimable et non imprimé, met en cause les modalités de lecture classique. Jean-Pierre Balpe est un grand spécialiste du leurre, mais d'un leurre au second degré : le simulacre est devenu simulacre de simulacre. À l'inverse, il adopte la position post-moderne classique de l'effacement de

l'auteur : ce dernier disparaît de la scène du texte car ce qu'il écrit, le programme, est hors d'atteinte du lecteur. Autrement dit, Balpe considère la délimitation qui passe au milieu du dispositif technologique dans la figure 2 comme la frontière entre le domaine de la production et celui de la réception, un décalage en somme de la frontière de la figure 1. En revanche, le rôle du dispositif technologique demeure un impensé dans sa conception : il continue à envisager un rapport causal entre le programme source et le résultat produit à l'exécution, c'est-à-dire que la génération automatique se fonde encore sur une conception potentielle du texte généré : « [le programme source] est un autre substitut qui transmet une pensée qui dit ». ⁶¹ Ce faisant, même si l'acte d'écriture est différé dans le programme, il ne change pas de fonction : le sujet humain en situation d'auteur continue à imprimer son intentionnalité par délégation dans le programme, même si celle-ci devient plus abstraite et ne s'exprime plus qu'à travers des niveaux conceptuels et non de surface. Ce qui est modifié, au fond, dans l'écriture, est la séparation entre la conception et l'inscription : si le sujet humain auteur est seul en situation de gérer, par la programmation, l'activité du « vouloir », c'est-à-dire la conception de « l'œuvre », se positionnant ainsi en situation fonctionnelle de méta-auteur, l'ordinateur, via le processus d'exécution, est, lui, positionné en situation de scribe : il a mission d'inscrire « l'œuvre ».

Le Méta-auteur ne peut qu'abdiquer une des parties importantes des responsabilités que lui assignait jusque là la tradition littéraire : il ne peut qu'être mis à distance du texte terminal dont il ne maîtrise pas toutes les composantes. [] Il met une distance entre l'écriture et la subjectivité. ⁶²

Cette théorie, dite du méta-auteur, résout la question du « robot poète », dans le sens, d'ailleurs, qu'avait pressenti le débat de 1965 et déplace la dimension inadmissible de cette littérature sur le lecteur :

L'enjeu existentiel du texte se déplace de l'espace de l'auteur à celui lecteur [] Qu'il y ait, ou non dans l'arrière-plan, comme bruit de fond vaguement perceptible un auteur, plus exactement qu'il y ait un auteur d'un programme de génération n'a, en dernière analyse, plus beaucoup d'importance : ce qui profondément importe, c'est l'instauration d'un rapport radicalement nouveau à la lecture [...] Ici, aucun alibi crédible d'enjeu

existentiel, toute littérature informatique, quel que soit l'angle sous lequel on l'examine, affirme la vacuité de l'auteur.⁶³

le lecteur est invité à faire preuve de liberté et à couper le « cordon » ombilical qui le lie à l'auteur dans sa recherche de « l'intentio auctoris » au sens d'Umberto Eco :

*L'intentio lectoris n'a plus à retrouver les indices de celle de l'intentio auctoris, mais se contente, d'une part, d'accepter que le texte lui parle et, d'autre part, de construire par ses actes un sous-ensemble particulier d'un roman général. Elle en acquiert une liberté réelle.*⁶⁴

Dans ce Lecteur Modèle, les textes susceptibles d'être générés ne sont donc pas tous posés comme équivalents, seules les solutions actualisées comptent : la conception potentielle est de fait remplacée au niveau du Lecteur Modèle par une conception virtuelle au sens de Deleuze et Lévy grâce à l'interactivité :

Le virtuel ne s'oppose pas au réel mais à l'actuel [í] L'actualisation est création, [í] il s'y passe autre chose que la dotation de réalité à un possible ou qu'un choix parmi un ensemble prédéterminé : une production de qualités nouvelles.⁶⁵

Alors que pour le méta-auteur, tous les textes susceptibles d'être générés demeurent équivalents : le méta-auteur demeure dans une conception potentielle de l'œuvre, ce qui fera d'ailleurs dire à Balpe :

Ce qui m'intéresse dans la génération automatique, ce n'est pas le texte qui s'affiche. Ce texte-là est un moment comme un autre, on s'en fout [í] Ce qui m'intéresse, c'est cette capacité à produire comme ça à l'infini et à générer un univers que je ne suis pas capable de faire. C'est donc un substitut qui transmet une pensée qui dit.⁶⁶

Toute la dimension de leurre de la génération automatique repose sur ce hiatus entre le potentiel et le virtuel instaurée entre le méta-auteur (Auteur Modèle) et le Lecteur Modèle.

c) La clôture de la génération combinatoire

Dans les années 1990, les démarches génératives de Pedro Barbosa et Christophe Petchanatz établissent un pont entre la génération et les autres formes développées dans cette période, notamment avec la génération de poésie animée.

Pedro Barbosa, met au point au CETIC à Porto le programme Syntext⁶⁷ à partir de 1992. Il s'agit d'un interpréteur, générateur de générateurs combinatoires qui fonctionnent sur le modèle classique des phrases à trous. L'auteur est amené à créer deux fichiers : le premier renferme la logique du générateur de texte, à savoir les modèles de phrases et les listes de vocabulaires, mais l'autre contient les instructions sur la façon dont le texte généré s'affiche à l'écran. Ce second fichier met ainsi sur le même plan l'algorithme générateur du matériau linguistique et celui qui génère la mise en forme de ce matériau à l'écran, dimension des textes qui sera au cœur de l'approche de la poésie animée. L'existence de ce second fichier interdit de résumer l'œuvre au seul algorithme génératif de sa dimension linguistique. Le second apport de Syntext est le modèle de l'écrit-lecteur qui lui sert de philosophie. Ce modèle stipule que le lecteur est co-auteur de ce qu'il lit par le jeu de l'interactivité, conception qu'on trouve également dans l'hypertexte de fiction, mais en plus, il établit une distinction explicite entre un champ des possibles qui ne s'est pas réalisé à la lecture et les textes qui ont effectivement été générés à la lecture. Syntext repose donc sur une approche virtuelle du texte généré, l'écrit-lecture constituant l'intervention créative qui permet d'actualiser le texte généré virtuellement présent dans les données interprétées.

Dans la pratique, l'interactivité permise par Syntext sera de fait assez limitée, relativisant très fortement cette conception virtuelle.

Christophe Petchanatz, quant à lui, fabrique au début des années 1990 des générateurs qui prennent à rebours certains dogmes de la génération. Il programme notamment des générateurs qui génèrent en continu des flux de textes à une vitesse telle qu'ils en deviennent illisibles et qu'il faut mettre le programme en pause pour lire. Le texte généré, ici, n'est plus un texte à lire, il se présente comme une information qui s'écoule sans que le lecteur ne puisse l'interpréter, une information à laquelle il ne peut accéder qu'en zapping, une information qui revendique ainsi sa dimension de production machinique vaine, « réflexe » et dont la structure lue n'est plus donnée par une formule abstraite

comme en génération automatique, mais par un arrêt programmé au niveau du système d'exploitation (la touche pause), donc, non par le méta-auteur, mais par l'entité technique qui a produit la machine et son système d'exploitation, positionnant ceux-ci comme co-auteurs du texte proposé à la lecture. Ces générateurs affirment également que le texte généré, avant même d'être un objet linguistique donné à la lecture, est un état transitoire observable inscrit dans le temps. Pour la première fois dans l'histoire de la génération, un auteur s'attaque à la conception du texte qui, dans les diverses démarches de génération que nous avons présentées garde une forme compatible avec la conception du texte imprimé et qu'on pourrait énoncer comme suit : « le texte généré est un objet purement linguistique qu'on peut extraire de son support, quel qu'il soit, pour le fixer dans un livre : c'est un texte imprimable ». Tout ceci est balayé chez Christophe Petchanatz, mais nous sommes au début des années 1990 et les conceptions de l'hypertexte et de l'animation s'étaient déjà développées.

6. Conclusion

La génération de texte a travaillé les problématiques de l'auteur, de l'œuvre et de la lecture, problématiques qui dépassent largement le domaine littéraire. Elle a proposé plusieurs points de vue sur ces questions. Toutefois, elle ne prend pas en compte l'ensemble des propriétés physiques du dispositif qu'elle utilise et garde quelques impensés. Ne donnant de signification qu'aux seuls algorithmes qui permettent de fabriquer un matériau textuel sans tenir compte de ceux qui permettent de l'inscrire à l'écran ni de ceux qui gèrent la relation avec la machine, ces points de vue sont incapables d'intégrer la labilité et supposent une relation totalement causale entre le programme source et le résultat obtenu à l'exécution. Cette relation ne subsiste malheureusement pas dès lors que la temporalité joue un rôle dans l'affichage, ce qui ne peut manquer d'apparaître, même sur les générateurs automatiques qui privilégient la création de textes statiques comme les générateurs automatiques de Jean-Pierre Balpe des années 1990. Elle ne pense pas non plus le rôle de l'interaction avec le lecteur et néglige toute dimension corporelle. Elle s'avère donc incapable de penser la génération dans les productions actuelles qui sont, pour bon nombre, des productions génératives, animées et interactives fondées sur le traitement d'un flux d'information. La génération a peu à peu évolué, on ne saurait aujourd'hui la résumer à la création d'un matériau linguistique. Ainsi donc, la « génération de

texte » est tout à la fois une approche technique qui perdure et une conception historiquement datée qui, pour moi, trouve sa clôture au début des années 1990 avec les démarches de Barbosa et Petchanatz. Elle sera néanmoins intégrée dans une forme plus générale et plus ouverte de littérature programmée définie comme une littérature multimédia programmée par un auteur et exécutée en temps réel lors de la lecture. Cette conception inclut la génération de texte, l'animation programmée et la création multimédia interactive programmée. Elle s'exprimera notamment dans le collectif *Transitoire Observable* (2003-2007).⁶⁸

NOTES

¹ Voir à ce sujet l'article de Beat Suter dans ce volume.

² Funkhouser, Chris. *Prehistoric digital poetry an archaeology of forms, 1959-1995*, Tuscaloosa : University of Alabama Press, 2007.

³ Le premier poème animé connu est *deux mots* de Roger Laufer, animé en images de synthèse par Michel Bret. Il s'agit d'un Film 16mm de durée 6' créé en 1982 et que Michel Bret présente comme « Images (et jeux) de mots sur un texte de Roger Laufer ». Il s'agit d'un texte réalisé en images de synthèses sur un ordinateur PDP 11/400 à l'Université Paris 8 et présenté à Cerisy-la-Salle en 1985 lors du colloque « La génération automatique de textes littéraires » (Laufer, Roger, « Les enfants du micro », Jean-Pierre Balpe, *L'imagination informatique de la Littérature*, Jean-Pierre Balpe & Bernard Magné eds., Saint-Denis : PUV, 1991 : 93-104).

⁴ Joyce, Michael. *Afternoon, a Story*, Cambridge : Eastgate Systems, 1987.

⁵ Date d'apparition du Kindle aux USA. Le Kindle ne sera commercialisé en Europe qu'à partir de 2009.

⁶ Par exemple : Funkhouser, Chris. *Prehistoric digital poetry an archaeology of forms, 1959-1995*, Tuscaloosa : University of Alabama Press, 2007. *Ibid.*, ou Bootz, Philippe. *Les Basiques : la littérature numérique*, Paris : Olats, 2007 [en ligne]

<<http://www.olats.org/livresetudes/basiques/litteraturenumerique/basiquesLN.php>> (consulté le 03/03/2014) ou encore le dossier de la BPI Bootz, Philippe. *La littérature numérique*, Paris : BPI, 2013 [en ligne]

<http://www.bpi.fr/fr/les_dossiers/culture_numerique/la_litterature_numerique.html> (consulté le 03/03/2014).

⁷ Turing, Alan Mathison. « On Computable Numbers, with an Application to the Entscheidungsproblem », *Proceedings of the London Mathematical Society*, (II/42, 1936) : 230-265. [en ligne]

<http://www.thocp.net/biographies/papers/turing_oncomputablenumbers_1936.pdf> (consulté le 25/02/2014).

⁸ Des étudiants de l'ENS Lyon en ont construit une version totalement mécanique en lego en 2012 à l'occasion du centenaire d'Alan Turing <http://www.dailymotion.com/video/xrn0yi_la-machine-de-turing-realisee_tech> (consulté le 25/02/2014).

⁹ Le programme source est le programme tel qu'il apparaît dans le langage de

programmation (C, Java, Flashí), c'est-à-dire la forme que l'homme peut comprendre. Ce programme n'est pas écrit en binaire mais dans un langage proche de l'anglais. Il n'est pas exécutable. Pour qu'il produise un résultat, il doit être soit compilé (transformé en programme exécutable autonome : les .exe sur un PC par exemple), soit interprété par un player qui va compiler ses instructions au moment de les exécuter (langage javascript dans une page Web par exemple).

¹⁰ Voir à ce sujet les exemples donnés dans Bootz Philippe. « La lisibilité du texte électronique », *L'encyclopédie de la recherche littéraire assistée par ordinateur*, Ottawa : L'Étalon [en ligne] (consulté le 03/03/2014).

¹¹ Lévy, Pierre. *Qu'est-ce que le virtuel ?* Paris : La découverte, 1998.

¹² Von Neumann, John. *First Draft of a Report on the ERVAC*, Philadelphia : University of Pennsylvania, 30 juin 1945.

¹³ Voir à ce sujet l'article Wikipedia sur le Manchester Mark I <http://fr.wikipedia.org/wiki/Manchester_Mark_I> (consulté le 10/02/2014).

¹⁴ Leavitt, David. *The man Who Knew Too Much: Alan Turing and the Invention of the Computer*, London: Phoenix 2007: 236, traduit en Français dans l'article de Wikipedia relatif au Manchester Mark I.

¹⁵ Turing, Alan M. « Computing machinery and intelligence », *Mind*, (59, 1950): 433-460. [en ligne] <<http://www.loebner.net/Prizef/TuringArticle.html>> (consulté le 10/02/2014).

¹⁶ Celui-ci débute en 1959 par un article célèbre : Lettvin, J.Y., Maturana, H.R., Mc Culloch, W.S., & Pitts, W.H., « What the Frog's Eye Tells the Frog's Brain », *Proceedings of the IRE*, (Vol. 47, No. 11, 1959): 1940-51.

¹⁷ Marquis, Pierre, Papini, Odile et Prade, Henri. « *Machines pensantes* » *d'hier, aujourd'hui et demain*, Inria, 2012. [en ligne] <<http://www.cnrs.fr/ins2i/IMG/pdf/prade-2.pdf>> (consulté le 10/02/2014).

¹⁸ Strachey, Christopher. « The 'thinking' machine », *Encounter* (13, October 1954) : 25-31. [en ligne] <<http://www.unz.org/Pub/Encounter-1954oct-00025>> (consulté le 10/02/2014).

¹⁹ « [í] even if the process seems fantastically complicated and laborious » ; Strachey, Christopher. « The 'thinking' machine », *Encounter* (13, October 1954). *Ibid.*, 25.

²⁰ « without it a computer is like a typewriter without a typist or a piano without a pianist » ; Strachey, Christopher. « The 'thinking' machine », *Encounter* (13, October 1954). *Ibid.*, 25.

- ²¹ Strachey, Christopher. « The 'thinking' machine », *Encounter* (13, October 1954). *Ibid.*, 27.
- ²² Couffignal, Louis, Pierre. *Les machines à penser*, Paris : Éditions de Minuit, 1952.
- ²³ Ducrocq, Albert. *L'ère des robots*, Paris : Julliard, 1953.
- ²⁴ Vian, Boris. « Un robot-poète ne nous fait pas peur », *Arts* (10-16, avril 1953): 219-226.
- ²⁵ Vian, Boris. « Un robot-poète ne nous fait pas peur », *Arts* (10-16, avril 1953). *Ibid.*, 220.
- ²⁶ Link, David. « There Must Be an Angel. On the Beginnings of the Arithmetics of Rays » in Siegfried Zielinski, and David Link, *Variantology 2. On Deep Time Relations of Arts, Sciences and Technologies*, Cologne: König, 2006 : 15-42, 25.
- ²⁷ Dates citées par David Link dans Link, David. « There Must Be an Angel. On the Beginnings of the Arithmetics of Rays », Siegfried Zielinski, and David Link eds., *Variantology 2. On Deep Time Relations of Arts, Sciences and Technologies*, Cologne: König, 2006. *Ibid.*, 15.
- ²⁸ Extrait cité dans Wardrip-Fruin, Noah. « Christopher Strachey: The first digital artist? », *Grand Text Auto*, August 1, 2005. [en ligne] <<http://grandtextauto.org/2005/08/01/christopher-strachey-first-digital-artist/>> (consulté le 12/02/2014).
- ²⁹ Ducrocq, Albert. *L'ère des robots*, Paris : Julliard, 1953. *Ibid.*, 265-266.
- ³⁰ Ducrocq, Albert. *L'ère des robots*, Paris : Julliard, 1953. *Ibid.*, 267.
- ³¹ Caillois, Roger. Ulam, Stanslaw, Monod, Jacques, De Ajuriaguerra, Julian, Calegero, Guido, Niel, Henri, Vercors, Ansermet, Ernest. *Rencontres Internationales de Genève 1965 : Le Robot, La Bête et l'Homme*, Neuchâtel : Éditions de la Baconnière, 1965 : 240-241. [en ligne] <<http://dx.doi.org/doi:10.1522/030078806>> (consulté le 20/02/2014).
- ³² Caillois, Roger, Ulam, Stanslaw, Monod, Jacques, De Ajuriaguerra, Julian, Calegero, Guido, Niel, Henri, Vercors, Ansermet, Ernest. *Rencontres Internationales de Genève 1965 : Le Robot, La Bête et l'Homme*, Neuchâtel : Éditions de la Baconnière, 1965, *Ibid.*, 237-253.
- ³³ Paul Eluard. « L'objectivité poétique », dans *Le surréalisme au service de la révolution* (numéro 5, 1933) : 13-14. [en ligne] <http://melusine.univ-paris3.fr/Surr_au_service_dela_Rev/Surr_Service_Rev5.htm> (consulté le 18/02/2014).
- ³⁴ Caillois, Roger. Ulam, Stanslaw, Monod, Jacques, De Ajuriaguerra, Julian,

Calegero, Guido, Niel, Henri, Vercors, Ansermet, Ernest, *Rencontres Internationales de Genève 1965 : Le Robot, La Bête et l'Homme*, Neuchâtel : éditions de la Baconnière, 1965. *Ibid.*, 238.

³⁵ Robinet, André. *Le défi cybernétique l'automate et la pensée*, Paris : Gallimard, 1973, 12.

³⁶ Caillois, Roger, Ulam, Stanislaw, Monod, Jacques, De Ajuriaguerra, Julian, Calegero, Guido, Niel, Henri, Vercors, Ansermet, Ernest. *Rencontres Internationales de Genève 1965 : Le Robot, La Bête et l'Homme*, Neuchâtel : Éditions de la Baconnière, 1965. *Ibid.*, 243.

³⁷ Caillois, Roger, Ulam, Stanislaw, Monod, Jacques, De Ajuriaguerra, Julian, Calegero, Guido, Niel, Henri, Vercors, Ansermet, Ernest. *Rencontres Internationales de Genève 1965 : Le Robot, La Bête et l'Homme*, Neuchâtel : éditions de la Baconnière, 1965. *Ibid.*, 247.

³⁸ Caillois, Roger, Ulam, Stanislaw, Monod, Jacques, De Ajuriaguerra, Julian, Calegero, Guido, Niel, Henri, Vercors, Ansermet, Ernest. *Rencontres Internationales de Genève 1965 : Le Robot, La Bête et l'Homme*, Neuchâtel : éditions de la Baconnière, 1965. *Ibid.*, 248.

³⁹ Lutz, Theo. « Stochastische Texte », *augenblick* (4, H. 1, S. 3-9, 1959). (consulté le 10/02/2014) ; version anglaise (en ligne, consulté le 10/02/2014).

⁴⁰ Ce générateur de textes voulait illustrer l'algorithme de génération de nombres aléatoires.

⁴¹ Intervention de M. Aurel David, in Caillois, Roger, Ulam, Stanislaw, Monod, Jacques, De Ajuriaguerra, Julian, Calegero, Guido, Niel, Henri, Vercors, Ansermet, Ernest. *Rencontres Internationales de Genève 1965 : Le Robot, La Bête et l'Homme*, Neuchâtel : éditions de la Baconnière, 1965. *Ibid.*, 242.

⁴² Caillois, Roger. Ulam, Stanislaw, Monod, Jacques, De Ajuriaguerra, Julian, Calegero, Guido. Niel, Henri, Vercors, Ansermet, Ernest. *Rencontres Internationales de Genève 1965 : Le Robot, La Bête et l'Homme*, Neuchâtel : Éditions de la Baconnière, 1965., *Ibid.*, 251.

⁴³ Lenoble, Michel. « Entre évanescence et fixité : le statut de la littérature générée par ordinateur », *A : LITTÉRATURE ↯*, Bootz, Philippe ed., Villeneuve d'Ascq : MOTS-VOIR, GERICO-CIRCAV, Université de Lille 3, 1994 : 26-28.

⁴⁴ Baudot, Jean. *La machine à écrire, mise en marche en marche et programmée par Jean A. Baudot*, Montréal : Les Éditions du Jour, 1964.

⁴⁵ Pour une présentation de la pièce, voir Vaïs, Michel. « Équation pour un homme actuel de Pierre Moretti » *Jeu : revue de théâtre*, (105 (4), 2002) : 120-

121. [en ligne] < <http://id.erudit.org/iderudit/26280ac> > (consulté le 26/02/2014).
- ⁴⁶ Chiffre avancé par Lenoble Michel in Lenoble, Michel. « Entre évanescence et fixité : le statut de la littérature générée par ordinateur », *A:\ LITTERATURE ↵, Les Cahiers du CIRCAV*, Bootz Philippe ed., Villeneuve d'Ascq : MOTS-VOIR, GERICO-CIRCAV, université de Lille 3, 1994. *Ibid.*, 28.
- ⁴⁷ Chiffre avancé par Pierre Moretti in Moretti, Pierre. « Note sur Équation pour un homme actuel », *Théâtre et Université* (n° 13, mars-avril 1968), 61.
- ⁴⁸ Funkhouser, Chris. *Prehistoric digital poetry an archaeology of forms, 1959-1995*, Tuscaloosa : University of Alabama Press, 2007. *Ibid.*
- ⁴⁹ Barras, Ambroise. « Quantité / Qualité. Trois points de vue sur les générateurs automatiques de textes littéraires », *Frontenac* (14, 1997-98) : 6-15. [en ligne] <<http://hypermedia.univ-paris8.fr/jean/infolit/Barras.html#t4>> (consulté le 10/02/2014).
- ⁵⁰ Bailey, Richard W.. *Computers Poems*, Michigan : Protagonising Press, 1973.
- ⁵¹ <http://www.ouliipo.net/> (consulté le 01/03/2014).
- ⁵² Deleuze, Gilles. *Différence et Répétition*, Paris : PUF, 1968, 169-176.
- ⁵³ Lévy, Pierre. *Qu'est-ce que le virtuel ?*, Paris : La Découverte, 1998. *Ibid.*, 14.
- ⁵⁴ <http://www.alamo.free.fr/pmwiki.php> (consulté le 01/03/2014).
- ⁵⁵ Balpe, Jean-Pierre, « Prélude », *Action poétique 95 ALAMO*, (2° trimestre 1984), 3.
- ⁵⁶ (en ligne, consulté le 01/03/2014).
- ⁵⁷ Le terme « privé » signifie que ce schéma et les problématiques qu'il soulève concernent essentiellement les situations dans lesquelles le sujet lecteur accède à l'œuvre depuis son ordinateur personnel, ce qui est la situation dans laquelle s'est majoritairement développé la littérature sur écran, notamment grâce aux éditions sur disquettes et cédérom.
- ⁵⁸ Balpe, Jean-pierre. *A:\ LITTERATURE ↵, Les Cahiers du CIRCAV*, Bootz Philippe ed., Villeneuve d'Ascq : MOTS-VOIR, GERICO-CIRCAV, Université de Lille 3, 1994. *Ibid.*, 24.
- ⁵⁹ Balpe, Jean-Pierre, « poème 8 » *Hôtel Continental* (n° X, 1981).
- ⁶⁰ Balpe, Jean-Pierre et Baboni-Schilingi, Jacopo. « Génération automatique poésie-musique » *Rencontres-médias* (1 1996-1997), Balpe, Jean-Pierre, coord., Paris : BPI Centre Pompidou, 1997, 149.

⁶¹ Balpe, Jean-pierre. *A:\ LITTERATURE ↵, Les Cahiers du CIRCAV*, Bootz Philippe, ed., Villeneuve d'Ascq : MOTS-VOIR, GERICO-CIRCAV, Université de Lille 3, 1994. *Ibid.*, 36.

⁶² Balpe Jean-Pierre. « Méta-auteur », *alire10/DOC(K)S*, 1997, 98.

⁶³ Balpe, Jean-Pierre. « Pour une littérature informatique : un manifestéi », *Littérature et informatique la littérature générée par ordinateur*, Vuillemin, Alain et Lenoble, Michel (coords.), Arras : Artois Presses Université, 1995, 23.

⁶⁴ Balpe Jean-Pierre. « Un roman inachevé-Dispositifs » *Littérature* (96, 1994), 52

⁶⁵ Lévy, Pierre. *Qu'est-ce que le virtuel ?* Paris : La Découverte, 1998. *Ibid.*, 13-15.

⁶⁶ Balpe, Jean-pierre. *A:\ LITTERATURE ↵, Les Cahiers du CIRCAV*, Bootz Philippe ed., Villeneuve d'Ascq : MOTS-VOIR, GERICO-CIRCAV, Université de Lille 3, 1994. *Ibid.*, 36.

⁶⁷ Acronyme de « synthétiseur de textes ».

⁶⁸ <http://transitoireobs.free.fr/to/> (consulté le 01/03/2014)

