

TEXTE ET PROGRAMME

Michel BERNARD

VOICI DEUX TEXTES

TEXTE 1

Sous le pont Mirabeau coule la Seine
 Et nos amours
 Faut-il qu'il m'en souviene
 La joie venait toujours après la peine
 Vienne la nuit sonne l'heure
 Les jours s'en vont je demeure
 Les mains dans les mains restons face à face
 Tandis que sous
 Le pont de nos bras passe
 Des éternels regards l'onde si lasse
 Vienne la nuit sonne l'heure
 Les jours s'en vont je demeure

TEXTE 2

```
PROGRAM anagrammes;
USES crt;
VAR
mot : STRING;
FUNCTION anagramme :STRING;
VAR
i,hasard : BYTE;
car : CHAR;
```

```

BEGIN
  RANDOMIZE;
  FOR i:=1 TO LENGTH (mot) DO
  BEGIN
    hasard:=RANDOM (LENGTH(mot)-1) + 1;
    car:=mot[i];           {Inversion de la ième lettre}
    mot[i]:=mot [hasard];  {de mot et d'une autre   }
    mot [hasard] := car;   {tirée au hasard     }
  END;
  anagramme:=mot;
END;

BEGIN
  CLRSCR;
  GOTOXY (1,12);
  WRITE ('Mot à traiter: ');
  READ (mot);
  REPEAT WRITELN (anagramme) UNTIL KEYPRESSED;
END.

```

Le premier est un poème bien connu d'Apollinaire, le deuxième est un programme en Turbo-Pascal¹. La simple juxtaposition de ces deux textes pourra passer auprès de certains pour de la provocation. Mon propos est cependant de montrer que les différences sont ici bien moins significatives que les points communs.

Commençons, *pianissimo*, par quelques constats d'évidence qui auront le mérite de ne choquer personne et d'entrer en douceur dans le vif du sujet. Ces deux textes présentent toutes les caractéristiques de la communication, telle qu'elle est analysée par Jakobson² :

1) ÉMETTEUR

Ces deux textes ont été écrits par des êtres humains. La personnalité du scripteur n'a ici aucune importance. Le poème

1 Turbo-Pascal est une marque déposée par Borland.

2 R. Jakobson, Essais de linguistique générale.

d'Apollinaire pourrait très bien être un plagiat ou un travail collectif, le texte du programme est anonyme, comme le sont par ailleurs beaucoup d'œuvres littéraires célèbres. Dans les deux cas, l'écriture est un acte conscient de communication. Par ailleurs, une machine ne peut écrire ni un poème ni un programme informatique : elle ne pourrait, au mieux, que combiner des morceaux de poèmes ou de programme (c'est ce que font par exemple les générateurs de programme).

2) RÉCEPTEUR

Ces deux textes sont écrits pour être lus par des êtres humains. Le fait que le texte 2 puisse avoir un sens pour une machine n'est pas essentiel : la machine, après compilation ³, ne réagira qu'à une série de 0 et de 1. Ce programme-source est écrit, règle d'or de la programmation, pour être lisible. On y trouve par exemple des remarques (encadrées ici par des { }) ignorées par la compilation et destinées aux lecteurs du programme. Les mots-clés de Turbo-Pascal sont en majuscules et les lignes sont indentées, conventions dont la machine n'a que faire mais qui facilitent la compréhension du texte.

3) MESSAGE

Ces deux textes visent à produire un effet par la communication d'un message. Il s'agit dans un cas de l'effet poétique, immédiat ou construit, causé par la lecture du texte et l'association d'images textuelles. Dans le cas du programme informatique, l'effet est obtenu par la médiation de l'ordinateur. Dans un cas comme dans l'autre, c'est la communication d'une information qui est première. Ecrire un programme informatique, c'est faire faire une tâche par l'ordinateur mais c'est aussi, le plus souvent, faire dire un message par la machine à son utilisateur. Le programme ci-dessus, par exemple, a pour objet d'afficher à l'écran une suite d'anagrammes aléatoires.

³ Opération effectuée par un programme spécialisé et consistant en une traduction du programme-source (écrit en langage évolué) en un programme-objet (écrit en langage-machine), directement exécutable par l'ordinateur.

4) CODE

Ces deux textes utilisent un vocabulaire et une syntaxe qui leur préexistent. Que le langage mis en œuvre soit une langue naturelle ou un langage artificiel, la compréhension du message passe forcément par l'existence d'un code, commun à l'émetteur et au destinataire. Le poème d'Apollinaire, comme le programme informatique, obéit à des règles (syntaxiques, prosodiques...). Le code imposé par Turbo-Pascal est celui d'un langage informatique dit "de haut niveau", c'est à dire compréhensible à la fois par la machine et par l'homme.

5) CANAL

La transition de ces deux textes d'un émetteur à un récepteur suppose l'existence d'un canal de communication. Remarquons ici que la référence à l'informatique ne change en rien le problème. Le programme doit figurer sur un support lisible par l'ordinateur, certes, mais ce peut être également le cas du poème. Par ailleurs, on peut très bien écrire un programme sur le papier sans jamais le transférer sur une surface magnétique (pour des raisons pédagogiques, par exemple).

6) RÉFÉRENT

La compréhension de chacun de ces deux textes, au delà de la simple lecture, suppose un corps de connaissances commun à l'émetteur et au destinataire. J'ai beau connaître le français, je dois savoir ce que sont le pont Mirabeau et la Seine pour comprendre le poème d'Apollinaire. Je dois savoir, pour bien lire le programme en Turbo-Pascal, ce que sont les UNITs dans la version 4 de ce langage, je dois connaître la définition d'un écran d'ordinateur pour apprécier l'effet de l'instruction GOTOXY.

Ces constatations sont élémentaires mais elles permettent d'asseoir le premier principe : écrire un programme est un acte de communication comme un autre, à savoir qu'il ne s'agit pas seulement de communiquer avec une machine mais avec des êtres humains.

On peut envisager l'écriture d'un programme informatique sur le modèle du presse-bouton : il n'y a pas de différence fondamentale entre le fait d'appuyer sur un interrupteur pour allumer une lampe et celui d'écrire un programme informatique, l'ordinateur digital étant lui-même

conçu comme un extraordinaire complexe d'interrupteurs microscopiques⁴. Il n'y a donc pas, dans ce sens, de communication : je ne peux pas dire que je communique à l'ampoule mon désir de la voir s'allumer !

Par contre, en actionnant l'interrupteur, je communique avec mes voisins, qui apprennent ainsi que je suis rentré chez moi, avec l'EDF qui me facturera ma consommation ; une lampe qui s'allume pourra signifier "Il est tard", "Silence on enregistre", "Je suis là", ou même, métaphoriquement, "Que la Lumière soit"... Nous retrouvons les constatations faites plus haut : le programmeur ne communique pas avec la machine mais avec d'autres hommes. Quels que soient les mécanismes contenus dans la "boîte noire", il y a toujours des hommes en amont et en aval⁵.

Le texte du programme informatique est à lire dans cette perspective. Il ne diffère pas essentiellement du poème, qui nécessite lui aussi toutes sortes de truchements pour mettre en communication le poète et son lecteur : impression, édition, lecture à voix haute, mise en musique... On a affaire dans les deux cas à une communication différée. Sur la base de cette analogie, il nous est maintenant possible de prendre sur les deux textes quelques points de vue communs.

STYLE

L'écriture d'un texte, quel qu'il soit, fait intervenir les choix du scripteur. On parle par exemple, dans ce sens, de "style de programmation". Deux programmeurs, respectant le même cahier des charges et utilisant le même langage, n'écriront pas le même programme.

L'algorithme retenu peut être différent. Ainsi, dans le programme proposé plus haut, on aurait tout aussi bien pu constituer l'anagramme en remplissant une seconde chaîne de caractères avec les lettres de la première, tirées au hasard. La structure du programme, également, peut varier. La saisie du mot, par exemple, aurait pu faire l'objet d'un sous-programme.

Il est à noter également que les langages évolués comme Turbo-Pascal permettent de nommer les objets de programmation comme on le souhaite. Les termes "anagramme", "mot" ou "hasard" auraient pu être

⁴ La plus grande différence, mais qui n'a pas été prise en compte ici, est la possibilité de branchement conditionnel du programme informatique.

⁵ On trouvera une illustration littéraire de cette signification du texte informatique dans le dernier roman de U. Eco, *Le pendule de Foucault*.

différents. Le choix de l'identificateur "i" est le résultat d'une pratique courante chez les programmeurs (c'est l'initiale de "itération"). On trouve également dans le programme des remarques et des chaînes à afficher qui sont à analyser comme n'importe quel texte. Une remarque à cet égard : le texte d'Apollinaire pourrait très bien constituer la sortie sur papier d'un programme rédigé de la manière suivante :

```
PROGRAM mirabeau;
USES printer;
BEGIN
    WRITELN (LST,'Sous le pont Mirabeau coule la Seine');
    WRITELN (LST,'      Et nos amours');
    WRITELN (LST,'      Faut-il qu"il m"en souviennne');
    WRITELN (LST,'La joie venait toujours après la peine');
END.
```

TYPOGRAPHIE

On sait qu'Apollinaire supprima au moment de l'édition de son poème tous les signes de ponctuation. Le jeu des alinéas, des centrages, des sauts de ligne, des majuscules, est par ailleurs essentiel en poésie. L'importance est la même dans l'écriture d'un programme. L'indentation des lignes, les sauts de lignes, l'utilisation des majuscules et des minuscules sont faits pour améliorer la lecture du code.

RÈGLES

L'écriture d'un texte, que ce soit dans le cadre de la prosodie française ou dans l'environnement de Turbo-Pascal, se fait dans les limites de règles contraignantes qui sont un gage de lisibilité. De même que le programmeur est obligé de fermer chaque parenthèse qu'il a ouverte, le poète est obligé de respecter le système de rimes qu'il s'est fixé.

Entendons-nous sur le statut de cette "obligation" : il est bien évident que la langue naturelle, même dans le cadre plus strict de la poésie, laisse plus de liberté au scripteur que la langue artificielle (pas question pour le programmeur de supprimer toute ponctuation de son texte...). Cependant, il est non moins évident que cette liberté du poète

est limitée par les attentes du lecteur et la faculté de compréhension de celui-ci. Une poésie qui ne respecterait aucune règle ne serait plus un vecteur de communication. S'il y a dérapage par rapport à une norme, il est toujours contrôlé.

INTERTEXTUALITÉ

On s'est intéressé depuis longtemps aux rapports qu'entretenait le texte littéraire avec d'autres textes, préexistants. Cette problématique est tout aussi pertinente dans le cas du programme informatique. D'une part, le programme en langage évolué fait appel, continuellement, à d'autres programmes. La mention "USES crt", par exemple, indique que certains des mots-clés employés par la suite sont définis dans un autre programme. D'autre part, le programmeur opère en fonction d'une "culture informatique" acquise au cours de ses études et de ses activités professionnelles, qui lui donne des réflexes comparables à ceux du joueur d'échecs. C'est parce qu'il a déjà lu et écrit de nombreux programmes qu'il est en mesure, par analogie, de trouver intuitivement des réponses adaptées aux problèmes qu'il se pose. Le poète, l'écrivain en général, ne pratiquent pas autrement : ils sont lecteurs avant d'être scripteurs.

Quelles conclusions tirer de ces rapprochements ?

1°) La programmation des ordinateurs est une activité de communication parmi d'autres, et plus particulièrement un acte d'écriture. On écrit un programme autant pour d'autres hommes que pour la machine.

2°) Le texte d'un programme est à la fois :

- Performatif : faire faire une série d'opérations à la machine
- Interactif : communiquer un résultat à l'utilisateur par le biais d'un dialogue
- Explicatif : expliquer comment on est arrivé à ce résultat

3°) La programmation relève donc des mêmes techniques et des mêmes approches que la production d'autres textes. Le texte informatique, qu'un état primitif des techniques de programmation rapprochait des frustes énoncés du langage mathématique, est devenu comparable aux textes en langage naturel.

Les conséquences de ce changement de point de vue sont notables dans le domaine du recrutement et de la formation des informaticiens,

qui ont été d'abord des électroniciens, puis des cybernéticiens, et enfin des mathématiciens.

Les programmeurs du XXI^e siècle seront linguistes.

Michel BERNARD
Professeur agrégé de Lettres Modernes
Lycée Paul Eluard
93000 Saint Denis