



Département
Éducation
et Technologie

- Numérisation
- Traitement formel
- Ordinateur ET logiciel

1. L'outil informatique

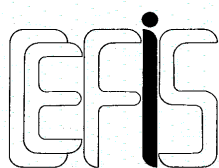
Etienne Vandeput

Basé sur la formation des personnes ressources en 1998 – 1999

Formateurs : Monique Colinet, Colette Coton, Rupert Meurice de Dormale

5.63

Février 1999



Centre pour la Formation à
l'Informatique dans le Secondaire

Chacun des documents de cette série est construit sur un modèle identique. Il s'agit, à propos du thème développé, de clarifier les notions et de mettre en évidence les concepts incontournables qui lui sont liés, dans l'optique de la préparation d'une formation.

La liste de ces concepts et notions ne constitue en rien une séquence rigide qui servirait de modèle de formation. Elle a seulement l'ambition de souligner les éléments qu'un enseignant, formateur de ses collègues, devrait avoir dans la tête lorsqu'il a l'intention de leur apprendre des choses à propos des technologies de l'information et de la communication..

1. L'outil informatique

1.1 Introduction

L'idée qui prévaut est qu'un nombre non négligeable de problèmes rencontrés par les utilisateurs dans leur pratique des TIC (technologies de l'information et de la communication) pourraient être résolus par eux, si la représentation qu'ils ont du système "ordinateur - programmes" est corrigée, pour se rapprocher davantage du mode de fonctionnement réel de ce système.

En particulier, la connaissance d'un minimum de choses à propos de la manière dont est traitée l'information (sous les formes que nous lui connaissons): sa numérisation, sa codification, sa transformation et sa reproduction sous une forme que nous (re)connaissons, nous paraît fondamentale dans la (re)construction de cette représentation.

Ce point de vue est loin d'être partagé par tout le monde, et certainement pas par ceux dont les préoccupations sont davantage de vendre du matériel et des logiciels et d'entretenir des marchés. Des acteurs de l'enseignement pensent aussi qu'il n'y a rien à dire à propos des TIC, qu'il suffit d'utiliser. Si vous êtes de cet avis, le type de réflexion qui suit ne vous intéressera pas. Nous persistons à croire que ces principes sont à rappeler au cours de la résolution de nombreux problèmes rencontrés par les utilisateurs, dans l'idée de corriger la représentation qu'ils se font de ce que nous avons appelé "l'outil informatique".

Ce thème est donc général. Les concepts et les principes évoqués se retrouvent et sont d'application dans tous les domaines d'activité auxquels les TIC s'appliquent, qu'ils soient passés, actuels ou à venir.

1.2 Idées maîtresses

Trois aspects du traitement de l'information sont à mettre en évidence. Ils sont fondamentaux et leur mauvaise compréhension empêche les utilisateurs de résoudre un nombre considérable des problèmes rencontrés au cours d'une session de travail. Nous les énonçons de la manière suivante:



Pour être traitable par un "ordinateur", toute information doit être numérisée.



Pour être possible, tout traitement d'une information par un "ordinateur" doit être décrit de manière purement formelle.



Ce n'est pas à l'ordinateur que l'utilisateur est confronté, mais au couple indissociable "ordinateur-logiciel", un exécutant sans cesse en évolution au cours d'une même session de travail.

Ces principes justifient que le mot *ordinateur* ait été écrit entre guillemets. Ils donnent un premier éclairage de ce qu'est un ordinateur.



Un ordinateur est une machine à traiter des informations de manière formelle pour autant qu'on lui ait préalablement indiqué comment mener à bien ce traitement.

Ch. Duchâteau

La bonne compréhension de ces principes est de nature à augmenter l'autonomie des utilisateurs:

- ils devraient appréhender de manière plus correcte l'évolution technologique tant logicielle que matérielle
- ils devraient être capables d'anticipation concernant les capacités attendues des programmes
- ils devraient manifester des capacités d'adaptation appréciables en ce qui concerne le passage d'une version d'un logiciel à une autre, d'un logiciel spécifique à un autre du même type, d'un type de produit à un autre type, voire d'un type d'ordinateur à un autre.

1.3 Suggestions pour un développement de ces idées

Les trois principes sont liés.

Le principe de la numérisation est justifié par les caractéristiques matérielles de l'exécutant. Les circuits électroniques, en tant que supports de l'information, imposent que celle-ci soit représentable sous forme binaire. Les nombres sont des symboles tout désignés pour cette représentation.

Le principe du traitement formel est lié, en partie, au fait que la numérisation fait perdre tout sens à l'information.

Quant à la connaissance des exécutants, elle sera d'autant mieux anticipée que les deux principes qui précèdent auront été assimilés.

1.3.1 La numérisation de l'information

L'ordinateur est construit sur le principe des circuits électroniques. Cette énorme contrainte impose que tout ce qui lui est fourni, qu'il s'agisse de commandes, de programmes ou de données à traiter, doivent être transformés en nombres (et même ramenés aux deux symboles 0 et 1, mais ça, c'est de l'ordre du détail). Cette numérisation a des conséquences gigantesques sur le traitement de l'information, notamment le fait que tout traitement doit être formalisé (cfr 1.3.2) pour pouvoir être exécuté. Ceci nous permet de rappeler la première assertion:



Pour être traitable par un ordinateur, une information (telle que nous la connaissons) doit être numérisée.

Une des conséquences importantes est qu'il faut des dispositifs pour numériser les informations et des dispositifs pour nous la rendre sous une forme acceptable (cfr 2.1).

Il faut définir des systèmes de codages des informations qui vont permettre de les traiter. Une partie de l'énergie des informaticiens consiste à trouver des manières de coder, des façons de numériser l'information. Une autre partie de cette énergie consiste à rechercher des codages qui facilitent la description des traitements escomptés. Si on approfondit la question, on se rend compte qu'une manière de coder n'est jamais vraiment complètement indépendante des algorithmes de traitement qui sont développés et inversement. Lire à ce propos: [Initiation à l'informatique Ch.3](#) et [Codage et traitement formel](#).

Les informations qu'on a pu à ce jour et progressivement numériser sont essentiellement:

- les nombres
- les textes
- les images
- les sons

Il n'est pas inutile de comprendre comment on y est arrivé, car les traces de ces démarches sont encore bien présentes dans l'utilisation quotidienne des programmes. Citons à titre d'exemples les expressions suivantes souvent rencontrées dans les menus de commande de programmes courants: *Texte ASCII(DOS)*, *Bitmap 24 bits*, *Qualité CD*,...

Nous ne développons ici que les principes généraux. D'autres détails seront fournis dans les documents concernant la création multimédia.

Numérisation des textes

Il vaut mieux parler de caractères, puisqu'un texte est d'abord un ensemble de caractères. N'importe qui, et donc aussi l'utilisateur moyen, peut imaginer un système de codification des caractères en nombres. Parmi les solutions qui peuvent être proposées, il faut tout de même distinguer les codages de longueur fixe des codages de longueur variable.

Voici deux petits exemples de solutions qui peuvent être proposées, en se limitant à coder les lettres majuscules et en se passant des caractères particuliers tels les signes de ponctuation.

Exemple 1

A=1; B=2; ... Z=26

Il est difficile de décoder une telle suite de nombres sans ambiguïté. On a besoin d'un délimiteur de lettres, voire de mots.

Par exemple, on décide de ne jamais utiliser ni le 0 ni le 9

A=1; B=2; ... H=8; I= 11; J=12; ...Z=32

et on choisit 0 comme séparateur de lettres et 9 comme séparateur de mots.

Dans ce cas, 1 2 0 5 9 2 6 0 1 7 0 2 5 0 2 3 9 2 3 0 1 0 1 4 0 2 5 0 5 se traduit sans hésitation par JE VOUS SALUE. On pourrait d'ailleurs sans trop de problèmes décrire l'algorithme de décodage.

Exemple 2

On voit que ce n'est déjà pas tout simple. Une autre solution consiste à décider de coder tous les caractères avec deux chiffres. On peut de la sorte, repérer plus facilement les lettres et même les mots si on décide d'un code pour l'espace.

A=01; B=02; ... Z=26; espace=00

Le message devient 1 0|0 5|0 0|2 2|1 5|2 1|1 9|0 0|1 9|0 1|1 2|2 1|0 5

A partir de ces deux exemples, le principe du codage des textes est établi. On observera qu'avec deux chiffres on peut coder cent caractères différents. Chacun jugera de l'opportunité d'aller plus loin dans les précisions concernant les codes véritablement adoptés: EBCDIC, ASCII, ANSI,... Au besoin, on peut faire sans peine le lien avec l'octet et les 256 caractères qu'il permet de coder. Mais l'essentiel est de percevoir, à travers cette nécessité de numériser, que les traitements ne seront pas faciles à décrire et que certains traitements souhaités ne pourront tout simplement pas être décrits.

Numérisation des images

Le problème est un peu plus épineux. Mais à bien y réfléchir, il n'est pas impossible, même à un novice, de découvrir une manière de procéder.

Celle qui revient volontiers est la technique, aujourd'hui encore la plus employée, du tramage de l'image à numériser. La zone à numériser est découpée en rectangles de dimensions égales (des pixels) et la couleur de chacun des pixels est codée. Bien entendu, le nombre de couleurs codées est limité (il en était d'ailleurs de même pour les caractères) et la finesse du tramage a aussi ses limites.

La découverte de ce processus peut conduire à quelques constatations importantes:

- la procédure est coûteuse en quantité de nombres produits lors du codage;
- la qualité du dessin rendu est fonction de la finesse de la trame;
- la taille de l'image codée est directement proportionnelle au nombre de couleurs codées et à la finesse de la trame (résolution).

Les images produites avec de telles techniques sont dites *images bitmap*. Par nature, les images affichées à l'écran, les images imprimées sont des images bitmap.

Cela dit, on souhaite souvent conserver des images sous une forme numérisée. C'est d'ailleurs tout l'intérêt d'un traitement informatique. Dans cette perspective, cette technique de codage

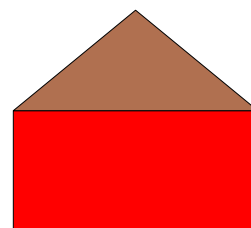
s'avère parfois peu pratique. On imagine donc d'autres techniques qui viseraient à considérer qu'une image n'est pas seulement un ensemble de pixels (comme l'imaginent les photographes, les peintres impressionnistes,...) mais comme un ensemble de formes possédant certaines caractéristiques faciles à numériser (comme l'imaginent plus volontiers les scientifiques, les techniciens et même certains artistes).

Ces autres techniques sont donc basées sur une description de l'image par des formules et des données qui permettent à l'ordinateur de la reconstituer en image bitmap lors de l'affichage et/ou de l'impression, avec l'avantage d'une mémorisation moins coûteuse en place.

En voici un exemple caricatural où cela vaut la peine. Le dessin se compose:

- d'un rectangle de couleur rouge
- d'un triangle de couleur brune

Les données à mémoriser sont peu nombreuses (longueur des côtés, position, mesure des angles, couleurs de remplissage...) eût égard à ce qu'elles seraient si cette image était considérée comme un ensemble de pixels (surtout si la résolution est élevée).



Ces techniques sont connues sous le nom de codage vectoriel. On parle d'images vectorisées.

Suite à la découverte de ces deux types de techniques de codage, on se rend aisément compte de la difficulté à traduire un paysage par des formules et de l'intérêt d'utiliser un codage bitmap. On perçoit comme il est pénible de considérer un schéma comme un ensemble de pixels et de l'intérêt d'utiliser un codage vectoriel.

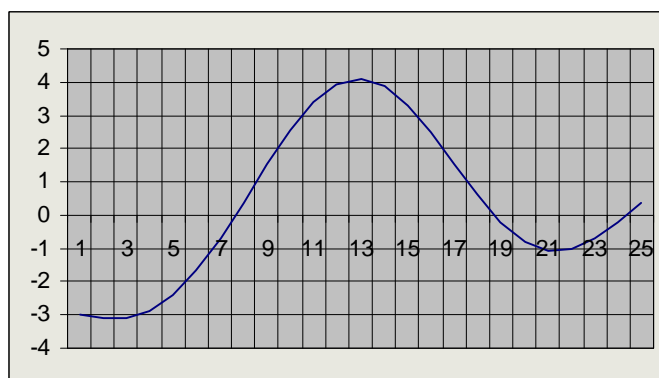
Dès à présent, on peut faire un certain nombre d'observations concernant des choses auxquelles tous les utilisateurs sont un jour ou l'autre confrontés. Citons-en quelques-unes:

- L'écran et l'imprimante produisent des restitutions de l'information qui peuvent différer compte tenu de leurs résolutions différentes.
- Scanner une image, c'est utiliser un procédé de numérisation basé sur le codage bitmap.
- Un programme de dessin technique, manipulant des images vectorisées, offrira des possibilités de traitement différentes d'un programme de retouche d'images ou de photos travaillant sur des images bitmap.
- Les deux types de codages et leurs variantes vont générer de nombreux formats d'image qu'il faudra apprendre à reconnaître avec leurs avantages et leurs inconvénients.

Numérisation des sons

Depuis quelques années, les ordinateurs sont en mesure de traiter les sons. Dans la logique qui vient d'être développée, cela signifie que des techniques de codage des sons ont été mises au point. Comme pour les dessins, il est utile de savoir lesquelles et d'en connaître les caractéristiques.

On a sans doute moins idée de ce qu'il serait possible de faire. Toutefois, une comparaison avec les images peut être faite. Si on sait qu'un son est le résultat d'une vibration et qu'on peut le modéliser sous forme d'une onde, une découpe de cette onde est également possible: découpe de l'amplitude et découpe dans le temps.



Les découpages horizontaux et verticaux se font évidemment de manière extrêmement fine. A titre d'exemple, le découpage dans le temps se fait plusieurs milliers de fois par seconde et le découpage de l'amplitude de l'onde se fait aujourd'hui en plus de soixante mille niveaux différents. C'est dire si dans ce cas, comme dans celui de l'image, les qualités rendues sont proches, et parfois supérieures à ce que nos propres sens peuvent capter.

Numérisation de ...

En combinant l'image et le son, on peut numériser des animations, des clips vidéo. Peut-être arrivera-t-on à numériser d'autres informations que du son, de l'image et du texte. Il existe d'ailleurs des dispositifs capables de détecter la modification de nombreuses grandeurs physiques: une pression, une température, une densité, une intensité lumineuse...

L'essentiel n'est-il pas de comprendre que des choses sont encore ou seulement possibles aujourd'hui. Le matériel, les programmes, les domaines d'activité touchés par les TIC changeront sans doute beaucoup dans l'avenir, mais les principes qui viennent d'être évoqués changeront peu ou pas. De leur bonne perception par les utilisateurs dépend en partie leur autonomie à gérer les difficultés qui se présentent en cours d'utilisation.

1.3.2 Le traitement formel

Il est une conséquence inévitable de ce qui précède. Les informations étant numérisées, c'est sur des nombres que travaille un ordinateur qui reste donc essentiellement un calculateur. Il importe d'en montrer la portée à ceux qui exploitent les TIC. Plusieurs pistes permettent de mettre en évidence cet aspect. En voici quelques-unes. Lire aussi: [Initiation à l'informatique Ch2](#).

Quand l'humain fait du formel...

Traiter des informations de manière formelle, c'est les traiter sans pouvoir leur accorder de sens. Un être humain ne le fait pratiquement jamais. Un ordinateur (au sens où nous l'avons défini) le fait toujours. On peut imaginer qu'un être humain effectue des traitements formels lorsqu'il manipule des informations dans une langue qu'il ne connaît pas: compter le nombre de "mots" d'un texte, supprimer les "phrases" qui contiennent un "mot" donné, lister tous les "mots" et les trier par ordre alphabétique (pour autant que l'alphabet utilisé lui soit connu),... Il ne pourra pas résumer, compter les formes verbales, traduire... parce qu'il est pratiquement impossible de formaliser ces traitements.

Nous avons volontairement appliqué des guillemets à *mot* et à *phrase* pour bien faire comprendre que l'être humain étranger à la langue du texte n'en reconnaîtra que la forme et en a donc une perception très limitée. Il en est de même d'un programme de traitement de texte, par exemple. Cette perception peut être améliorée si l'être humain dispose d'un dictionnaire. Cependant, le sens sera toujours absent du traitement.

L'être humain effectue quand même des traitements formels à l'occasion: calculer, conjuguer, trier... Dans ces cas-là, le traitement se ramène à exécuter une marche à suivre (programme) qu'il a plus ou moins parfaitement intégrée. Rappelons-nous les séances interminables de calcul et de conjugaison dans l'enseignement fondamental.

Dans le domaine des TIC, l'homme s'évertue à écrire pour les ordinateurs des programmes qui reculent les frontières de la formalisation.

A propos de sens et de forme...

On peut dire que les quatre informations qui suivent sont interprétées de la même façon par un être humain (qui connaît la langue dans laquelle elles ont été écrites):

Ordinateur ORDINATEUR ordinateur ordinnateur

Pourtant, elles ont toutes une forme différente. Même un étranger les trouverait différentes. C'est encore bien pire dans l'exemple qui suit:

QUATRE 4 IV quatre

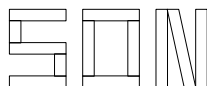
A l'opposé, des informations qui ont la même forme peuvent être interprétées de manières très différentes par un être humain qui peut lui accorder du sens:

Il a pris **son** mal en patience.

J'aime le **son** du cor...

Il faut séparer la farine du **son**.

This is my **son**.



Décrire un traitement formel...

On voit à quel point la contrainte de formalisation des traitements rend compliquée la description d'un traitement, si simple soit-il. De plus, lorsqu'une telle description a pu avoir lieu, elle garde une certaine rigidité qui ne peut disparaître que par une augmentation des instructions de traitement. Par exemple, il est possible de décrire la conjugaison des verbes en ER, en précisant la terminaison à ajouter au radical, à chaque personne, à chaque temps et à chaque mode.

Je chante Je chantais ...

Tu chantes Tu chantais...

... ..

On remarque vite que ça fonctionne mal avec les verbes *changer, nettoyer, lancer,...* Si donc on agrmente son explication de tout une série d'exceptions, celle-ci s'allongera considérablement. Elle n'est cependant pas encore complète dans la mesure où un étranger n'est pas capable de distinguer ce qui est un verbe de ce qui ne l'est pas (chanter, poirier, boiler, xyzer,...) Il faudra donc ajouter la liste de tous les verbes acceptables. On voit à quel point un traitement, même par nature assez formel, peut être complexe à décrire.

1.3.3 Le couple "ordinateur-logiciel"

Ceci nous amène tout naturellement à considérer que le logiciel est complètement lié au matériel. En d'autres mots, les contraintes matérielles sont telles que la description des traitements est souvent pénible et, pour certains traitements, souvent imparfaite, voire très éloignée de ce que l'on souhaite vraiment. Pensons à la correction grammaticale, telle qu'elle est aujourd'hui effectuée par les programmes de traitement de texte. Lire encore: [Initiation à l'informatique Ch4](#).

La conjonction du logiciel et du matériel transforme "l'ordinateur" en un exécutant (dispositif effectuant une marche-à-suivre). Le rôle de l'utilisateur est de le piloter. Cette manière de voir les choses transforme quelque peu la vision que peut avoir un utilisateur du travail avec un ordinateur: ce n'est pas l'utilisateur qui effectue le travail, même s'il intervient souvent; il ne fait que le piloter. Sa responsabilité se situe à ce niveau-là. D'autre part, une autre responsabilité est en jeu, celle qui se traduirait par des erreurs au niveau de l'écriture des programmes. Derrière le couple ordinateur-logiciel, se cache donc un troisième intervenant (même s'il semble absent du dialogue): le programmeur.

Il faut aussi observer que si, au cours d'une session de travail, le matériel ne change pas, le logiciel peut changer souvent. Aujourd'hui, les ordinateurs ne sont plus gérés par un seul mais par plusieurs programmes et les manipulations de l'utilisateur sont de nature à rendre actif, tantôt l'un, tantôt l'autre. La conversation de l'utilisateur avec l'exécutant se déroulant par l'intermédiaire de dispositifs standardisés (ou presque) tels un clavier, une souris,... la mission de pilotage n'en est que plus délicate. L'utilisateur n'est pas confronté à un mais à plusieurs exécutants. Reconnaître l'exécutant actif à un moment donné est donc fondamental car les dispositifs de commande (qui sont toujours les mêmes) provoquent des traitements différents en fonction des exécutants et des situations dans lesquelles ils se trouvent.

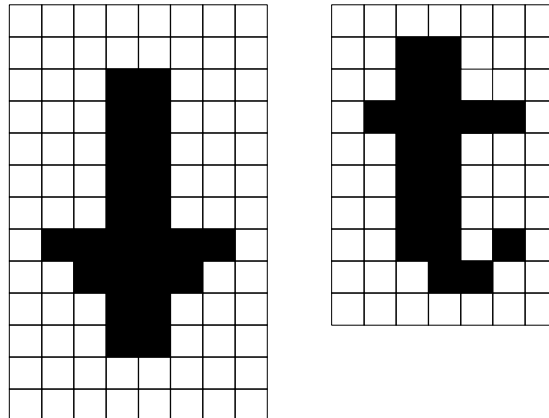
A titre d'illustration, on peut s'amuser à prévoir et à vérifier ce qui se passe lorsque telle touche du clavier est pressée, en envisageant le plus grand nombre de situations différentes.

Cet exercice est loin d'être inintéressant. Beaucoup de problèmes rencontrés par des utilisateurs peu exercés proviennent d'une mauvaise perception de la situation ou d'une mauvaise identification de l'exécutant. En voici un exemple à titre d'illustration.

Avec Windows 95, pour modifier le nom d'un document, il suffit de réaliser deux clics suffisamment espacés (quelques dixièmes de seconde) sur le nom du fichier dans la fenêtre de documents. Un double clic sur ce même nom doit en revanche provoquer l'ouverture du document. Si le double clic n'est pas correctement réalisé, il est interprété comme deux simples clics. Le système attend donc que l'utilisateur modifie le nom, ce qui surprend l'utilisateur. Un nouveau double clic désespéré de sa part restera sans effet.

Ceci dit, l'association matériel-logiciel souffre d'un handicap grave. Le fonctionnement matériel est basé sur l'utilisation de signaux électriques à travers des circuits électroniques. Cet état des choses impose que les programmes comme les données soient numérisés. Tant les informations que les instructions permettant de les traiter doivent être transformés sous l'unique forme acceptable par un ordinateur: des nombres. Nous en tirons la conclusion suivante:

Solutions:



4. *Quelle est la taille de l'image affichée à l'écran (800 pixels sur 600) enregistrée par l'intermédiaire d'un logiciel de dessin bitmap au format 24 bits.*

Solution:

Le codage de chaque couleur demande trois octets (24 bits). Il est à noter que le nombre de couleurs que l'on peut coder ainsi est supérieur à 16 millions.

La taille de l'image est donc: $800 * 600 * 3 = 1.440.000$ octets soit 1,37 Mo. Elle tiendrait à peine sur une disquette.

5. *Même question pour une image qui fait 200 pixels sur 150 si elle est enregistrée en 256 couleurs.*

Solution:

Dans ce cas, chaque couleur est codée sur un octet. Sa taille est de: $200 * 150 * 1 = 30.000$ octets soit 29,3 Ko (48 fois moins que dans l'exemple précédent).

6. *Vous avez enregistré votre voix pendant 2 minutes 15 et vous avez choisi la qualité CD comme qualité d'enregistrement. Quelle est la taille du fichier son qui a été généré?*

Solution:

La qualité CD correspond à une fréquence d'échantillonnage de 44,1 KHz (44.100 fois par seconde). Le niveau d'échantillonnage est de 16 bits (soit 2 octets ou encore 65536 valeurs possibles) et c'est en stéréo (2 pistes).

Dès lors, la taille sera de: $135 * 44.100 * 2 * 2 = 23.814.000$ octets soit 22,71 Mo.

7. *Même question si le fichier est enregistré en qualité téléphone.*

Solution:

La qualité téléphone correspond à une fréquence d'échantillonnage de 11,025 KHz, le niveau est de 8 bits (soit un octet) et une seule piste est utilisée.

La taille sera de: $135 * 11.025 * 1 * 1 = 1.488.375$ octets soit 1,42 Mo (soit 16 fois moins que le précédent).

1.4.2 Le traitement formel

1. Si on vous demande (à vous être humain) des anagrammes du mot CRANE, que proposerez-vous comme réponses? Que serait capable de faire un "ordinateur" face à la même situation?

Solution:

L'être humain ne peut en général se passer d'accorder du sens aux informations qu'il traite. Dans cette optique, les solutions proposées seront normalement: *ECRAN, NACRE, ENCRA...* tous des mots (au sens humain du terme) ayant un sens dans notre langue.

Le programme qui gère l'ordinateur auquel une telle question est posée peut très bien proposer toutes les permutations possibles des lettres du mot pour obtenir d'autres mots (qui risquent bien de n'avoir pas de sens pour un être humain). Il faut tout de même ajouter que si le programme est prévu pour gérer une liste de mots acceptables (un dictionnaire en quelque sorte), l'ordinateur pourra proposer les mêmes solutions que l'être humain, sans avoir toutefois attribué le moindre sens aux informations traitées.

2. Lorsque votre ordinateur est géré par votre programme habituel de traitement de texte, que considère-t-il comme des mots? Autrement dit, quelle définition formelle du mot doit être prise en considération pour cet exécutant?

Solution:

La réponse dépend évidemment du programme utilisé. Il faut cependant noter que d'une version à l'autre d'un même programme, cette définition peut changer. Les circonstances dans lesquelles le programme prend en considération le concept de mot sont, par exemple: le déplacement semi-automatique du point d'insertion (au début du mot suivant, du mot précédent), la sélection semi-automatique (du mot courant), la vérification orthographique, les statistiques (nombre de mots), l'espacement...

Une façon de procéder consiste à placer dans un texte des séquences particulières de caractères (chiffres ou caractères particuliers consécutifs ou non) et à observer comment s'effectuent le déplacement et la sélection semi-automatiques se rapportant au mot.

Vérifiez, par exemple, si les groupes de caractères qui suivent sont des mots pour votre traitement de texte. Tâchez de répondre avant d'essayer. Attention, le résultat n'est pas forcément identique en déplacement et en sélection.

12345azer
 chauve-souris
 l'informatique
 l'attrape-mouches
 123azer45
 azer,ty
 azer,,ty
 azlerty
 123
 1,1
 azer!!!ty)))uio-p"

3. *Quelle est la définition d'un mot pour un joueur de scrabble? Serait-il possible de programmer un tel jeu pour un ordinateur?*

Solution:

Les seuls mots acceptables au jeu de scrabble sont ceux du dictionnaire Larousse. On imagine qu'à l'occasion d'un concours, l'année d'édition est également précisée. Il est donc possible de programmer un jeu de scrabble qui valide et comptabilise les mots, à condition de permettre au programme la consultation du dictionnaire reconnu.

A remarquer aussi que l'utilisation par un joueur d'un mot valable ne l'oblige pas, sauf indication contraire d'un règlement, à en connaître la signification. Le jeu de scrabble a donc un côté formel assez prononcé.

4. *Parmi les traitements suivants, quels sont ceux qui sont faciles à formaliser, difficiles à formaliser, pratiquement impossibles à formaliser? Pour vous aider, imaginez que vous devez les effectuer vous-mêmes et que la langue du texte vous est inconnue.*

Compter le nombre de formes verbales dans un document

Supprimer dans un document toutes les parties de texte entre parenthèses

Traduire une notice explicative

Transcrire un texte manuscrit en texte imprimé

Transcrire un texte du présent au passé

Jouer aux cartes

Trouver quel jour est le ../../. sans avoir de calendrier sous la main

Dessiner une grille de mots croisés

Remplir une grille de mots croisés si les définitions sont fournies

Solution:

Il n'est pas impossible de détecter un certain nombre de formes verbales dans un texte, mais le traitement n'est pas à l'abri de dérapages (ne parlons pas d'erreurs puisqu'il s'agit de travailler de manière formelle). Comment en effet, un étranger pourrait-il, même en disposant d'un dictionnaire, reconnaître les formes verbales dans les cas suivants:

Il dit: "Je ne puis vous aider". Puis il s'en alla.

Les poules du couvent couvent.

Etre ou ne pas être, c'est la question que se pose tout être.

L'entrée A a posé des problèmes.

La reconnaissance d'une parenthèse ouvrante puis, plus loin, d'une parenthèse fermante ne fait pas intervenir le sens du texte. C'est donc un traitement qui peut être complètement formalisé. Vous pouvez tenter de le faire faire par un programme de traitement de texte.

Traduire une notice explicative est un traitement qui se heurte à la structure des langues. Si la traduction pouvait être littérale, le traitement pourrait être, en partie formalisé. Cependant, la traduction d'un mot est souvent différente suivant le sens qu'il a. Ce qui rend malgré tout le traitement difficile à formaliser. Comme la traduction littérale n'est pas possible, nous admettons que ce traitement est pratiquement impossible à formaliser, à l'exception de la traduction d'expressions toutes faites comme celles que l'on peut trouver dans de petits manuels ou traducteurs électroniques.

Transcrire un texte manuscrit en imprimé est possible dès l'instant où chaque caractère manuscrit peut être identifié sans ambiguïté à un caractère imprimé connu. C'est donc un traitement que l'on peut considérer comme formel à quelques petits problèmes près qui dépendront surtout de la qualité de l'écriture. À noter que si les symboles d'écriture utilisés sont trop étrangers à celui qui doit faire le travail, son travail risque d'être plus ardu.

Pour transcrire un texte du présent au passé, il faudrait pouvoir identifier de manière non équivoque les formes verbales, ce qui pose problème comme nous l'avons déjà dit. La conjugaison, en revanche, est un traitement purement formel si le radical est identifié et correspond bien à un verbe.

Le jeu de cartes et les jeux de stratégie en général peuvent être formalisés. La formalisation consiste essentiellement en la génération d'un nombre important de possibilités et, parmi celles-ci, le choix de celle qui statistiquement donnera le meilleur résultat. Ceci explique qu'il est possible de gagner contre un ordinateur mais aussi, qu'on puisse lui attribuer des niveaux d'expertise lorsque l'on joue contre lui.

Un calendrier peut être généré par une règle simple (du moins pour une période de plusieurs centaines d'années). En prenant comme point de départ, l'année 1900, une année sur quatre est bissextile, à l'exception des années dont le nombre est multiple de cent et n'est pas multiple de quatre cents. De la sorte, il doit vous être possible de fournir le jour de la semaine correspondant à n'importe quelle date du siècle prochain, même si ça vous prend un peu de temps.

D'une manière générale, tout dessin bien structuré peut être réalisé grâce à des consignes précises et un certain nombre de données: nombre de cases en ligne, en colonne, position des cases noires, dimension des cases... Plus généralement encore, on pourra confier la gestion d'un tableau à un exécutant formel en lui fournissant un minimum de consignes et de données.

La compréhension des définitions n'est guère sujette à formalisation. Elle l'est d'autant moins que les définitions des mots croisés peuvent être à double sens.

Ces différents exemples tentent de montrer les limites des traitements que l'on va pouvoir attendre, à court, à moyen et à long terme de la part d'un exécutant "ordinateur".

L'habileté de l'utilisateur à se servir de l'outil informatique va dépendre de son aptitude à découvrir de nombreux exécutants, dans des domaines d'applications assez différents, et qui ont comme point commun de ne pouvoir réaliser que des choses parfaitement ou imparfaitement formalisées. D'où l'importance de lui faire percevoir, au moment jugé le plus opportun, ces aspects fondamentaux du traitement de l'information.

Une dynamique efficace consiste à identifier les actions de base de ces exécutants par catégories (systèmes d'exploitation, traitements de texte, outils de création multimédia, outils de navigation,...) de manière à leur fournir une autonomie minimum. L'utilisateur pourra alors progresser dans sa connaissance des exécutants, à travers chaque problème qu'il doit résoudre avec lui. Cette progression ne peut avoir lieu s'il n'a pas bien conscience des principes qui viennent d'être développés.

Les documents qui suivent s'attacheront à décrire de grandes catégories de logiciels afin de mettre le doigt sur l'essentiel et de permettre une amélioration de la perception des TIC par leurs utilisateurs potentiels.

Prochain document: **2. Le matériel et le système d'exploitation**