Université Paris Ouest - Nanterre - la Défense

Licence de Sciences du langage
2008 - 2009

Informatique

Résumé de cours

Marcel Cori

Introduction

1 Pourquoi un enseignement d'informatique ?

L'informatique est un outil privilégié d'acquisition et de transmission des connaissances : toutes les connaissances sont stockées sur des supports informatiques, qui ont tendance à se substituer aux supports papier. De même, la circulation des connaissances passe à travers des canaux informatiques.

Mais l'utilisation bien à propos de l'outil informatique n'est pas très facile :

- il faut un apprentissage des logiciels ;
- les logiciels ne sont pas parfaitement au point, ils présentent parfois des erreurs ;
- il faut pouvoir se débrouiller en cas de pannes.

Enfin, quand on suit un enseignement supérieur, il faut avoir de la hauteur par rapport à l'outil informatique. Il ne suffit pas de savoir faire quelques opérations sur sa machine : il faut comprendre la logique de ce que l'on fait. Puisque il faudra utiliser cet outil de nombreuses années, il ne faut pas être attaché à la forme qu'il prend aujourd'hui, mais être en mesure de s'adapter à ses évolutions.

2 Pourquoi l'informatique intéresse spécialement les linguistes

1. Pour communiquer avec les ordinateurs, on utilise des *langages*. Il peut être intéressant de mettre au jour les différences entre ces langages et les langues humaines, que l'on appelle par oppositon des langues *naturelles*, d'étudier les particularités de la communication avec les ordinateurs.

2. Les données étudiées par les linguistes se trouvent sur des supports informatiques, que ce soient des données écrites ou des données orales. Les recherches en linguistique, par conséquent, nécessitent une utilisation spécifique de l'outil informatique.

Les linguistes utilisent plus particulièrement des *corpus* qui sont des collections de données langagières, collectées selon certaines procédures. Les corpus sont censés être des échantillons significatifs d'une langue donnée, ou de certains usages de la langue. On a accès à des corpus de taille de plus en plus grande. La constitution et l'exploitation des corpus sont des tâches que doivent maîtriser les linguistes.

3. On effectue des *traitements* sur les données langagières qui sont stockées sur les supports informatiques. On peut citer, de manière non exhaustive :

- la traduction automatique, ou l'aide à la traduction ;
- la correction orthographique ou grammaticale ;
- la reconnaissance de la parole ;
- la recherche d'information.

La réalisation de logiciels de *Traitement automatique des langues* (TAL) met en jeu des équipes ou figurent des informaticiens, mais aussi des linguistes susceptibles de communiquer avec des informaticiens. D'où des professions dans le domaine du Traitement automatique des langues (on parle aussi d'ingénierie linguistique), accessibles à des étudiants qui ont fait des études de linguistique[1].

[1] Il ne faut pas confondre l'*utilisation* des logiciels informatiques, qui est le fait de tous les linguistes, avec la *réalisation* de ces logiciels, qui implique une minorité d'entre eux.

1 Les ordinateurs et leur environnement

1.1 La structure fonctionnelle des ordinateurs

1.1.1 Vue générale

Un ordinateur est composé :
- d'un processeur ;
- d'une mémoire ;
- d'organes d'entrée ;
- d'organes de sortie.

1.1.2 La mémoire

La mémoire sert à stocker les programmes et les données.

On parle de mémoire *RAM* : *Random Access Memory*. Cela signifie accès *aléatoire*, autrement dit quel que soit l'élément de la mémoire auquel on accède, cela prend le même temps. L'accès aux éléments de la mémoire est rapide.

La mémoire est *volatile*. Cela signifie que si on coupe le courant toutes les informations contenues s'effacent. On parle aussi de *mémoire vive*.

Opposée à la mémoire RAM, il existe une mémoire *ROM* : *Read Only Memory*. On peut accéder aux informations stockées en ROM, mais on ne peut en enregistrer de nouvelles, ou du moins l'enregistrement est plus difficile et plus lent que l'accès.

La mémoire ROM de l'ordinateur contient des informations nécessaires au démarrage, qui seraient perdues à chaque fois qu'on coupe le courant si elles étaient en RAM. Entre autres :

- le programme qui permet de reconnaître les entrées/sorties principales (programme *BIOS*, *Basic Input/Output System*) ;
- le programme qui permet de charger le système d'exploitation (à partir du disque dur, d'un CD, d'une disquette).

L'accès aux informations de la mémoire ROM est moins rapide qu'aux informations de la RAM.

Outre la mémoire *centrale*, il existe des mémoires *périphériques*, qui sont en fait des supports extérieurs à l'ordinateur : les *disques durs*, les *disquettes*, les *CD-ROM* (*Compact Disc*), les *DVD-ROM* (*Digital Video Disc* ou *Digital Versatile Disc*), les *clés USB*.

Ces mémoires sont des mémoires *permanentes* : on ne perd pas les informations quand le courant est coupé. Leur coût est plus faible que celui des mémoires centrales, mais le temps d'accès aux informations est beaucoup plus élevé.

Les mémoires sont caractérisées par leur capacité (en *octets* ou en *bits*[2]), ainsi que par le temps d'accès aux informations. Les unités utilisées pour ces mesures sont les suivantes :

- les méga-octets : un méga-octet, noté 1 Mo, vaut un million d'octets, ou encore 10^6 octets ;

[2] Un octet correspond à huit bits. Voir ci-dessous page 16.

- les giga-octets : un giga-octet, noté 1 Go, vaut un milliard d'octets, ou encore 10^9 octets ;
- les micro-secondes : une micro-seconde, notée 1 μs, vaut un millionième de seconde, ou encore 10^{-6} seconde ;
- les nano-secondes : une nano-seconde, notée 1 ns, vaut un milliardième de seconde, ou encore 10^{-9} seconde.

Les ordres de grandeur des matériels qu'on trouve actuellement dans le commerce sont les suivants :
Mémoire RAM : de 128 à 2000 Mo
Disque dur : de 20 à 500 Go
CD-ROM : 700 Mo
Clé USB : de 2 à 64 Go
Disquette : 1,44 Mo
DVD-ROM : de 4,7 à 9,4 Go

1.1.3 Le processeur

Le *processeur* est encore appelé *CPU (Central Processing Unit)*. Le processeur est l'organe qui permet l'exécution des *instructions* :
- les *opérations arithmétiques et logiques* portant sur les données stockées dans la mémoire centrale ;
- les *tests* sur ces données, ainsi que les *opérations conditionnelles* ;
- les *transferts internes* de données ;
- les *entrées et sorties*.

Il existe des « cases » spécifiques de la mémoire, appelées *registres*, qui jouent un rôle privilégié dans ces différentes opérations.

C'est le processeur qui donne sa « personnalité » à l'ordinateur. Le premier microprocesseur date de 1971. Il a permis la réalisation des micro-ordinateurs. Des exemples actuels de (micro)-processeur sont le *PentiumD Dual Core* de la firme *Intel* ou l'*Athlon 64* de *AMD*.

Le processeur est cadencé au rythme d'une *horloge interne*. Le nombre d'impulsions par seconde se mesure en *Hertz*. On compte plutôt en *mégaHertz (MHz)* ou *gigaHertz (GHz)*.

1.1.4 Les organes d'entrée et de sortie

Les organes d'entrée et de sortie permettent à l'ordinateur de communiquer avec l'extérieur.

- Les *ports série* envoient des informations (bits[3]) les unes à la suite des autres. Les ports série bidirectionnels permettent d'envoyer et de recevoir des données.

- Les *ports parallèle* envoient simultanément 8 bits.

[3] Voir plus loin, page 12.

- Les *bus USB (Universal Serial Bus)* sont de type série, mais ils ont un fonctionnement beaucoup plus rapide que les ports série standard. Ils fournissent l'alimentation électrique aux périphériques auxquels ils sont reliés.

1.1.5 Les périphériques

1.1.5.1 Généralités

Les *périphériques* sont des matériels externes à l'ordinateur, qui communiquent avec l'ordinateur par l'intermédiaire des organes d'entrée et sortie. Les périphériques permettent :

- d'accéder à des informations externes à l'ordinateur (clavier, souris, scanner, lecteur de CD-ROM, de DVD-ROM, webcam) ;

- de recevoir des informations en provenance de l'ordinateur (moniteur, imprimante, amplificateur, graveur de CD ou de DVD, vidéoprojecteur) ;

- les deux (lecteur de disquette, disque dur, modem, lecteur/graveur).

Sans les périphériques les êtres humains ne pourraient se servir des ordinateurs.

1.1.5.2 Les moniteurs

Il y a deux types de moniteurs : les écrans à tube cathodique et les écrans plats. Les caractéristiques des moniteurs sont :

- la *taille* (de la diagonale) qui se mesure en pouces (*inches*). Beaucoup de moniteurs actuels ont une taille comprise entre 14 à 21 pouces.

- la *définition*, qui correspond au nombre de points qui peuvent être affichés, ou *pixels* (*picture elements*) : l'image de l'écran est formée par une matrice d'éléments graphiques.

1.1.5.3 Le modem

Le *modem* est un périphérique utilisé pour transférer des informations entre plusieurs ordinateurs, par exemple via les circuits téléphoniques. Les ordinateurs contiennent des informations numériques (sous forme binaire), les lignes téléphoniques fonctionnent de manière analogique.

Le modem *module* les informations numériques en ondes analogiques, il *démodule* les données analogiques pour les convertir en numérique.

La vitesse de transmission se mesure en *bauds*, ou en bits par seconde.

Remarque : la différence analogique/numérique

Des sons ou des images peuvent être enregistrés sur des supports physiques, selon deux modes : le mode analogique ou le mode numérique.

Dans le mode analogique, la réalité physique reçoit une représentation sous la forme d'un autre objet physique. Ce qui fait que quand on veut reproduire la représentation obtenue, il y a une perte. Il y a également une perte parce que l'objet physique qui sert de représentation s'altère avec le temps.

Dans le mode numérique, la représentation est informationnelle, cela veut dire qu'elle correspond à un codage. Ce qui fait que la reproduction, la transmission et la conservation peuvent en être parfaites.

1.1.6 La carte mère

La *carte mère* est une carte électronique qui possède des connecteurs pour le processeur, des barrettes de mémoire, des cartes d'extension :

- le *chipset* est un jeu de circuits électroniques chargé de coordonner les échanges de données entre les divers composants de l'ordinateur (processeur, mémoire) ;

- le *CMOS* (*Complementary Metal Oxide Semiconductor*) est un circuit constamment alimenté (par une pile ou une batterie) qui conserve des informations (heure, date, structure des disques durs) de manière permanente ;

- les *connecteurs d'extension* (*slots*) sont des réceptacles dans lesquels on peut insérer des cartes d'extension, qui permettent de compléter et d'améliorer les potentialités de l'ordinateur.

1.2 Programmes et données

Les programmes comme les données sont stockés dans les mémoires (centrales et périphériques) des ordinateurs.

1.2.1 Programmes

Les *programmes* agissent sur des *données (d'entrée)* pour produire de nouvelles *données (de sortie)*.

C'est l'*utilisateur* d'un programme qui entre les données. À chaque fois qu'on entre les mêmes données, le programme fournit les mêmes résultats. L'ordinateur applique bêtement, rapidement, de manière infaillible ce que le programme lui dit de faire.

Tout ce que fait un ordinateur est commandé par un programme. À un moment donné, plusieurs programmes entrent en jeu afin de faire fonctionner l'ordinateur.

C'est ainsi qu'un programme peut lui-même être la donnée d'entrée d'un autre programme. Les programmeurs écrivent leurs programmes dans des langages dits évolués (C, Java, Python, Pascal,...). Ces langages sont extrêmement rigoureux, mais leur syntaxe est compréhensible par des êtres humains. La séquence qui suit constitue par exemple un fragment de programme écrit dans le langage Python :

```
s = 1
i = 1
while (i < 11) :
     s = s*i
     i = i+1
```

Afin d'être *exécutés*, les programmes écrits dans des langages évolués doivent être traduits en *langage machine*, c'est-à-dire en un langage binaire, un langage constitué de suites de 0 et de 1. Cette traduction est effectuée à l'aide de programmes spécifiques : *compilateurs* ou *interpréteurs*. Un programme écrit dans le langage Pascal est ainsi la donnée d'entrée d'un compilateur, qui le traduit en un programme écrit en langage machine, ou programme *exécutable*[4].

[4]La description du processus est ici simplifiée.

1.2.2 Les systèmes d'exploitation

Les *systèmes d'exploitation* regroupent les programmes qui gèrent les ressources de l'ordinateur, afin que d'autres programmes puissent s'exécuter. Il s'agit de gérer l'affichage à l'écran, l'accès aux fichiers, l'enchaînement de l'exécution des programmes, etc.

Des exemples de systèmes d'exploitation sont *Unix*, *MS-DOS*, *Windows 98*, *Windows XP*, *Windows Vista*, *Linux*.

Il faut bien noter que le système d'exploitation est composé de programmes, il n'est pas partie intégrante de la machine, ce n'est que de l'« intelligence » qui est ajoutée à la machine. Un même ordinateur pourrait fonctionner avec des systèmes d'exploitation différents.

À chaque fois qu'on met un ordinateur sous tension, tout se passe comme s'il refaisait son « éducation » en chargeant les programmes de la mémoire ROM, puis les programmes du système d'exploitation qui proviennent en général du disque dur.

1.2.3 Les logiciels d'application

Les *logiciels d'application* sont destinés directement à l'utilisateur[5]. On peut citer *Word*, *Excel* et *PowerPoint* dans l'environnement *Microsoft*, *Acrobat Reader* produit par *Adobe*.

Remarque : On distingue les *logiciels libres* des *logiciels propriétaire*. Les logiciels libres ne sont pas nécessairement gratuits et les logiciels propriétaire ne sont pas nécessairement payants. Un logiciel libre peut être utilisé, recopié, comme on l'entend. Son « code source », c'est-à-dire son écriture en langage évolué, est connu, et donc il peut être étudié, modifié.

1.2.4 Fichiers

Un ensemble de données stockées sur une mémoire périphérique s'appelle un *fichier*. Un fichier peut contenir n'importe quel type de données, textes ou documents, sons, images, mais aussi des programmes.

Un fichier est repéré par son *nom*. Plus exactement, le nom peut se décomposer entre le nom proprement dit et une *extension* qui précise, en principe, la « nature » du fichier.

Ainsi, sous le système *Windows*, `Word.exe` désigne un programme (exécutable), alors que `lettre.doc` indique que l'on a affaire à un document, probablement créé par le logiciel *Word*; `poly.pdf` sera un document à lire sous *Acrobat*.

1.2.5 L'indépendance des programmes et des données

Un fichier peut avoir été créé par un programme, auquel cas il contient des données de sortie de ce programme. Les données contenues par un fichier peuvent également servir de données d'entrée d'un programme. Enfin, un même programme peut modifier le contenu d'un fichier, auquel cas les données du fichier sont alternativement des données d'entrée et des données de sortie.

[5]Le terme *logiciel* ne reçoit pas une définition univoque. Il peut désigner tous les programmes utilisés par un ordinateur ou un programme unique. En général, quand on parle d'un logiciel on parle essentiellement d'un programme auquel sont associés des ressources (ensemble de données regroupées dans des fichiers) ou d'autres programmes dépendants.

Un fichier de données est totalement indépendant du programme qui l'a créé. Ainsi, un fichier créé sous le *bloc-notes* de *Windows* peut être lu et modifié par *Word*. Etc.

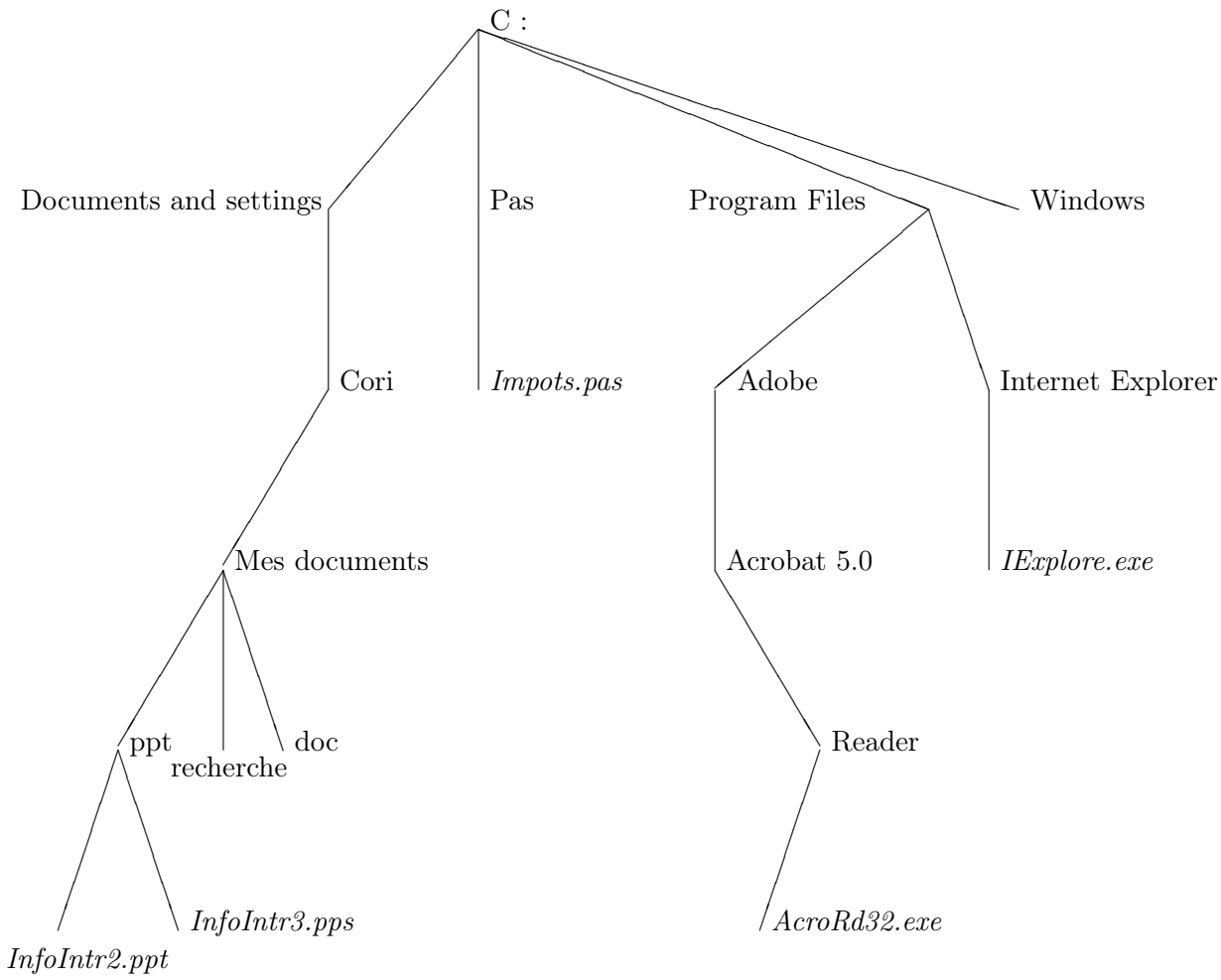
C'est le choix arbitraire de certains concepteurs de systèmes d'exploitation qui fait que quand on clique sur le nom d'un fichier, cela met en route un programme bien précis qui va prendre ce fichier comme donnée d'entrée.

1.2.6 L'organisation arborescente des fichiers

Les noms des fichiers qui sont stockés sur une mémoire périphérique donnée sont rangés dans le *répertoire* (*directory*) de cette mémoire périphérique. Tous les systèmes d'exploitation définissent des procédures permettant d'afficher le contenu d'un répertoire. Ainsi, sous *Windows* on peut se servir de l'« explorateur ». En *MS-DOS*,

```
dir C:
```

permet d'afficher le nom des fichiers du disque `C`.



Mais, afin d'avoir un classement rationnel des fichiers, les répertoires peuvent contenir des *sous-répertoires*, qui contiennent aussi des noms de fichiers. Et ainsi de suite, tout sous-répertoire pouvant contenir également de nouveaux sous-répertoires. Ce qui fait que l'on obtient une *arborescence*, comme celle très simple présentée ci-dessus, qui décrit (de manière très partielle) le contenu du disque dur (appelé C) d'un ordinateur. Les noms des fichiers sont en italique, afin d'être distingués des noms de sous-répertoires.

Dans l'environnement *Windows*, on parle de *dossiers* plutôt que de sous-répertoires, confondant (métaphoriquement) un système de classement avec une organisation physique : une mémoire périphérique contient des dossiers, lesquels contiennent des fichiers ainsi que des sous-dossiers, etc.

Il peut être utile de désigner un fichier par son nom accompagné de son *chemin d'accès*. Par exemple, en *MS-DOS*, on peut écrire :

```
C:\pas\impots.pas
C:\Documents and settings\Cori\Mes documents\ppt\InfoIntr3.pps
```

De cette façon, deux fichiers différents peuvent recevoir le même nom : il suffit que leurs chemins d'accès soient différents.

```
C:\Documents and settings\Cori\Mes documents
```

désigne un sous-répertoire (ou une sous-arborescence).

Sous *Unix*, les chemins sont décrits avec des *slash* plutôt qu'avec des *backslash*. Un fichier peut ainsi être désigné par :

```
pas/impots.pas
```

1.3 Internet et le Web

1.3.1 Les réseaux

Les ordinateurs peuvent communiquer entre eux. C'est-à-dire que les données de sortie d'un ordinateur peuvent servir de données d'entrée pour un autre ordinateur.

Les organes d'entrée/sortie en cause sont soit des *cartes réseaux* (par exemple la carte réseau *Ethernet*), soit des *modems* (pour la communication à distance *via* les lignes téléphoniques).

1.3.2 Internet

Internet (on dit aussi *l'Internet*) a été mis en place au début des années 1980 : il s'agissait de faire communiquer entre eux différents réseaux d'ordinateurs (*net* = *réseau*) situés à distance les uns des autres, en utilisant notamment les lignes téléphoniques.

Le langage commun de communication entre les ordinateurs est le protocole IP (*Internet Protocol*). La transmission des messages s'effectue par le protocole TCP/IP (transmission par paquets).

Sur Internet, chaque machine connectée a une *adresse IP* : c'est une suite de 4 nombres, chacun compris entre 0 et 255.

Par exemple : `195.83.48.68`

Les deux premiers nombres désignent une entreprise ou une organisation, le troisième nombre un réseau interne à l'entreprise, et le quatrième la machine proprement dite.

Mais il existe aussi une adresse plus facile à lire par des êtres humains :

`machine.u-paris10.fr`

Cette adresse indique clairement, dans l'ordre, d'abord le nom de la machine, puis le domaine où elle est implantée, enfin le pays[6]. Les adresses des machines situées aux États-Unis se terminent par des indications telles que `org`, `edu`, etc.

L'association des noms explicites aux adresses numériques se fait grâce à un système appelé *DNS (Domain Name System)*. Il existe des machines, appelées serveurs de noms, qui établissent le lien entre le nom et le domaine.

Il existe plusieurs applications d'Internet. Ainsi, le *courrier électronique* (*e-mail*) est une de ces applications[7]. Le *FTP (File Transfer Protocol)* est une autre application, qui permet le transfert de fichiers.

Le *Web*, ou *World Wide Web*, est une autre application. (On dit quelquefois *la Toile* en français). C'est toutefois le Web qui a donné sa popularité à Internet.

1.3.3 Le Web

Le premier serveur Web est apparu en 1991. L'idée directrice du Web est que des utilisateurs puissent consulter des informations situées sur des machines distantes et hétérogènes avec un confort quasi-équivalent à celui qu'ils auraient si ces informations étaient sur leurs machines personnelles.

Les informations sont composées à l'aide du langage HTML[8] (*Hyper Text Markup Language*). Elles utilisent le protocole HTTP.

Les informations sont situées sur des *pages* Web, qui ont chacune une adresse : l'*URL* (*Uniform Resource Locator*). Par exemple :

`http://infolang.u-paris10.fr/modyco/sommaire.htm`
`http://www.u-paris10.fr/linguist/licence.html`

Dans une telle adresse figurent :
- le protocole,
- la machine, puis le domaine où sont les données,
- un chemin d'accès (dans la syntaxe *Unix*),
- le fichier, avec une extension `html` ou `htm`.

`www` est le nom d'une machine dans un domaine donné, en général la machine qui fait office de « serveur » Web dans le domaine.

[6] On peut remarquer que si la lecture des adresses IP se fait de gauche à droite, celle des adresses explicites se fait de droite à gauche.

[7] En français on dit quelquefois *courriel*.

[8] Voir plus loin, page 19 et suivantes.

On accède à ces données grâce à un logiciel appelé *browser* : *Internet Explorer*, *Mozilla Firefox* ou *Netscape Navigator*. Nous dirons *logiciel de navigation*, ou *navigateur*.

Il devient possible d'afficher sur l'écran de son ordinateur personnel des données situées à des milliers de kilomètres de chez soi. Ces données auront été chargées au préalable dans la mémoire centrale de l'ordinateur personnel. Néanmoins, le navigateur peut sauvegarder provisoirement (éventuellement même pour plusieurs jours) les données sur le disque dur.

L'utilisateur peut également décider de sauvegarder, d'une manière permanente, les données qui l'intéressent. Il doit cependant se préoccuper du fait que sur l'écran peuvent être affichées des données provenant de plusieurs fichiers, et qu'en une seule opération de sauvegarde, il ne sauvegardera que l'un des fichiers.

2 La représentation informatique des données

2.1 Le codage binaire de l'information

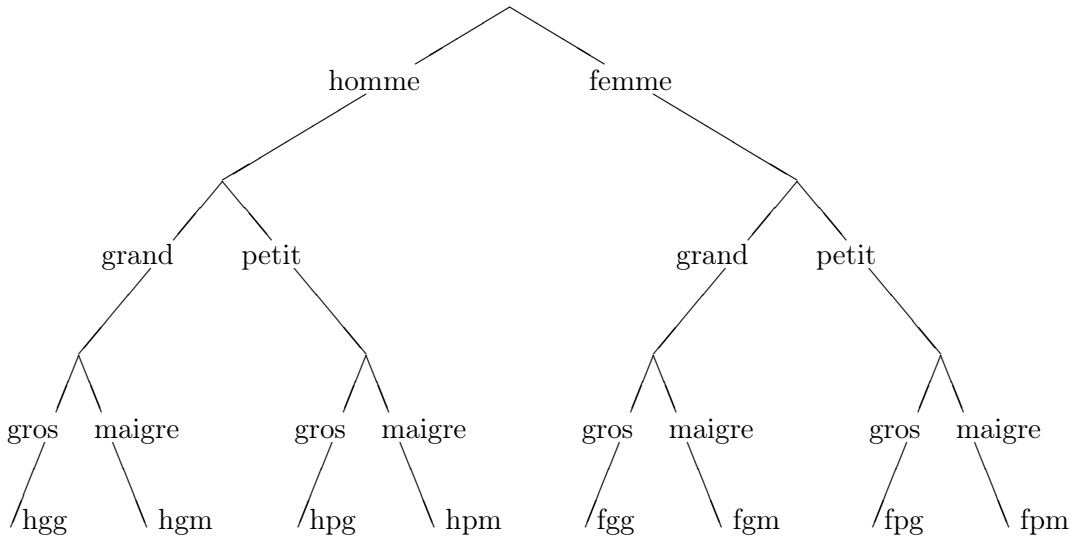
2.1.1 Des informations élémentaires aux informations complexes

Pour des raisons techniques (courant qui passe ou qui ne passe pas, interrupteur dans l'une ou l'autre position), les informations sont représentées sur les supports informatiques sous une forme *binaire*.

Le *bit* (*bi*nary dig*it*) est la quantité minimum d'information. Il correspond à une question à laquelle on peut répondre par *oui* ou *non*, ou par *vrai* ou *faux*.

Toute information peut être représentée par une suite de questions auxquelles on répond par *oui* ou par *non*, autrement dit par une suite de bits.

Ainsi, avec 3 bits on peut représenter 8 objets différents, ou 8 informations différentes, comme on le voit ci-dessous.

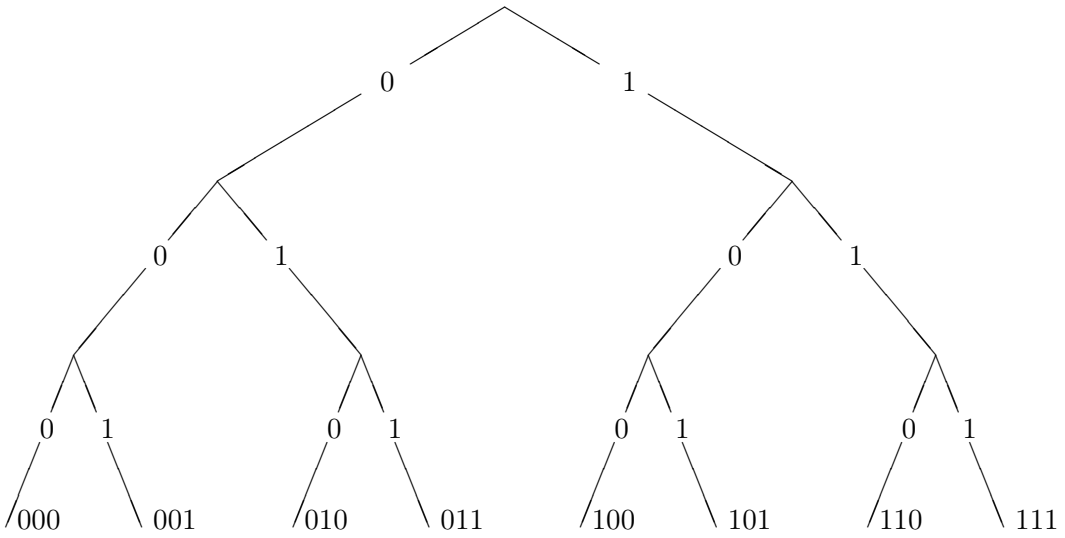


Avec 1 bit, on peut représenter 2 informations différentes ; avec 2 bits, on peut représenter $2 \times 2 = 4$ informations différentes ; avec 3 bits, on peut représenter $4 \times 2 = 8$ informations différentes ; avec 4 bits, on peut représenter $8 \times 2 = 16$ informations différentes, etc.

Plus généralement, avec N bits on peut représenter $2 \times 2 \times \cdots \times 2$ informations différentes, c'est-à-dire 2^N informations.

2.1.2 L'utilisation des nombres écrits en base 2

La représentation des deux valeurs possibles prises par une information s'effectue avec des 0 et des 1. Ce qui se traduit sous la forme arborescente suivante :



Le « chemin d'accès » à l'information obtenue à l'issue des trois choix peut s'écrire sous la forme d'une suite de `0` et de `1`. C'est donc un nombre écrit en base `2`, c'est-à-dire en binaire.

À chaque information complexe correspond ainsi un nombre, comme on le voit dans le tableau ci-dessous :

000	0	homme grand et gros
001	1	homme grand et maigre
010	2	homme petit et gros
011	3	homme petit et maigre
100	4	femme grande et grosse
101	5	femme grande et maigre
110	6	femme petite et grosse
111	7	femme petite et maigre

2.1.3 La traduction entre nombres binaires et nombres décimaux

Contrairement aux ordinateurs, nous manipulons plus facilement les nombres en base 10 (nombres décimaux) que les nombres écrits dans la base 2 (nombres binaires). C'est pourquoi il est utile de connaître des procédures permettant de passer des uns aux autres.

(1) Calcul de la valeur décimale d'un nombre écrit en binaire

En partant de la droite, on considère tous les `1`, et leur position dans la suite des chiffres. Ils correspondent respectivement aux valeurs décimales : `1, 2, 4, 8, 16, 32, ...`

Par exemple, `111` a pour valeur `1 + 2 + 4 = 7`, `1000` a pour valeur `8`, et `11010101` a pour valeur `1 + 4 + 16 + 64 + 128 = 213`.

(2) Écriture binaire d'un nombre décimal

On divise le nombre par `2`, puis on divise le quotient obtenu par `2`, et ainsi de suite. En écrivant de droite à gauche les restes obtenus au fur et à mesure, on obtient le nombre dont on

est parti écrit en base 2.

Par exemple :
37 divisé par 2 donne 18 et il reste 1 ;
18 divisé par 2 donne 9 et il reste 0 ;
9 divisé par 2 donne 4 et il reste 1 ;
4 divisé par 2 donne 2 et il reste 0 ;
2 divisé par 2 donne 1 et il reste 0 ;
1 divisé par 2 donne 0 et il reste 1.

37 s'écrit donc 100101 en base 2.

2.2 Le codage des caractères

2.2.1 Le problème

Pour coder les 26 lettres de l'alphabet, on a besoin de 5 bits ($2^5 = 32$). Si on veut distinguer les lettres majuscules des lettres minuscules, ce sont 6 bits qui deviennent nécessaires ($2^6 = 64$). Mais si on ajoute les 10 chiffres, ainsi que plus de deux signes de ponctuation, ou des symboles qu'on trouve sur les claviers tels que « + », « = », etc., il nous faut 7 bits ($2^7 = 128$).

2.2.2 Le code ASCII

Rien n'obligeait au départ les fabricants d'ordinateurs et les concepteurs de logiciels à adopter le même codage pour les caractères. Un organisme, l'ISO (*International Organization for Standardization*), a été constitué afin de proposer des normes.

C'est ainsi qu'a été défini le code ASCII (*American Standard Code for Information Interchange*), ou norme ISO 646.

Ce code utilise 7 bits, autrement dit les nombres qui vont de 0000000 (zéro) à 1111111 (127). Les principaux codes qui ont été définis par la suite, et qui utilisent plus que 7 bits, englobent le code ASCII.

Les 32 premiers nombres binaires, autrement dit les codes qui vont de 0000000 à 0011111, ne représentent pas à proprement parler des caractères, mais ce qu'on appelle des *caractères de contrôle*[9]. Les 96 codes[10] restants représentent les caractères ci-dessous :

[9] Nous donnons plus loin (page 17) un exemple d'utilisation des caractères de contrôle.

[10] On remarquera qu'on utilise le terme *code* à la fois pour désigner une convention de codage (le code ASCII) et pour désigner le nombre binaire qui représente un caractère donné (le code 0100001 qui représente le caractère A).

32 :	33 : !	34 : "	35 : #	36 : $	37 : %	38 : &	39 : ’
40 : (	41 :)	42 : *	43 : +	44 : ,	45 : -	46 : .	47 : /
48 : 0	49 : 1	50 : 2	51 : 3	52 : 4	53 : 5	54 : 6	55 : 7
56 : 8	57 : 9	58 : :	59 : ;	60 : <	61 : =	62 : >	63 : ?
64 : @	65 : A	66 : B	67 : C	68 : D	69 : E	70 : F	71 : G
72 : H	73 : I	74 : J	75 : K	76 : L	77 : M	78 : N	79 : O
80 : P	81 : Q	82 : R	83 : S	84 : T	85 : U	86 : V	87 : W
88 : X	89 : Y	90 : Z	91 : [	92 : \	93 :]	94 : ^	95 : _
96 : ‘	97 : a	98 : b	99 : c	100 : d	101 : e	102 : f	103 : g
104 : h	105 : i	106 : j	107 : k	108 : l	109 : m	110 : n	111 : o
112 : p	113 : q	114 : r	115 : s	116 : t	117 : u	118 : v	119 : w
120 : x	121 : y	122 : z	123 : {	124 : |	125 : }	126 : ~	

On notera que l’espace est un caractère comme un autre, dont le code est 32, c’est-à-dire 0100000.

On notera également que les chiffres, en tant que caractères, ont un code qui est différent du nombre qu’ils représentent. Par exemple le caractère 1 a pour code ASCII 49.

2.2.3 Les codes ultérieurs

Le code ASCII ne prend pas en compte les accents ou autres signes diacritiques qui sont utilisés en français ou dans d’autres langues. C’est pourquoi ont été définis les codes *ISO-LATIN* (norme ISO 8859), *ISO-LATIN-1* pour le français.

ISO-LATIN-1 est défini sur 8 bits, et permet par conséquent la représentation de 256 caractères.

La compatibilité avec le code ASCII se vérifie en ce sens qu’en supprimant le 0 par lequel commencent les 128 premiers codes, on obtient un code ASCII qui représente le même caractère. Par exemple, le caractère a est représenté par 01100001 en ISO-LATIN-1 ou 1100001 en ASCII, ce qui correspond toujours au nombre 97.

Dans le tableau ci-dessous sont indiqués certains des codages, qui concernent notamment les lettres accentuées.

192 : À	194 : Â	196 : Ä	198 : Æ	199 : Ç	200 : È	201 : É	202 : Ê
203 : Ë	206 : Î	207 : Ï	209 : Ñ	212 : Ô	217 : Ù	219 : Û	220 : Ü
224 : à	226 : â	230 : æ	231 : ç	232 : è	233 : é	234 : ê	235 : ë
238 : î	239 : ï	241 : ñ	244 : ô	249 : ù	251 : û	252 : ü	255 : ÿ

On remarque toutefois l’absence de codes pour les caractères œ, Œ et Ÿ.

Sous Windows, on utilise un autre code (CP 1252), qui a une compatibilité avec ISO-LATIN-1 mais n’est pas identique. Ce code permet notamment les représentations suivantes :

Œ	140
œ	156
Ÿ	159
€	128

Il existe aussi un code ISO 859-15, plus complet que le code ISO 8859-1.

En 1989 a été définie une nouvelle norme, la norme ISO 10646, qui, utilisant 32 bits, autorise le codage de plus de quatre milliards de caractères.

Un sous-ensemble de cette norme est le code *UNICODE* qui utilise 16 bits et autorise le codage de 65536 caractères. UNICODE est compatible avec ISO-LATIN-1.

2.2.4 Le codage des caractères non ASCII sur 7 bits

Il reste que tous les systèmes informatiques n'admettent pas de travailler sur 32, 16 ou même 8 bits. Ainsi, les transferts de données par Internet sous le protocole SMTP (*Simple Mail Transfert Protocol*) prennent en compte des caractères de 7 bits.

Il existe bien des procédures de codage/décodage entre 8 bits et 7 bits, comme la procédure MIME (*Multi-Purpose Internet Mail Extensions*), utilisée par des logiciels de courrier électronique.

Mais il est plus sûr pour éviter les risques de distorsions de ne transmettre que des caractères codés sur 7 bits (et donc n'utilisant que le code ASCII).

Les documents créés sous le format Latex ou le format HTML n'utilisent que les caractères du code ASCII. Ces documents ont par conséquent la propriété d'être plus que les autres indépendants des machines et des systèmes sur lesquels ils sont lus.

Les caractères non ASCII sont codés par une suite de caractères ASCII. Par exemple, le `é` est codé `\'e` en Latex et `&eacute;` en HTML.

2.3 Des fichiers aux documents

2.3.1 Fichiers textes

On peut considérer que les fichiers qui contiennent des données linguistiques sont constitués de suites de caractères. On les appellera *fichiers textes*[11]. On supposera dans ce qui suit que les fichiers textes sont formés de suites d'*octets* (*bytes*), un octet étant une suite de 8 bits.

Cela signifie que le fichier est organisé de manière linéaire, selon une dimension unique : on ne retrouve pas les deux dimensions de l'écran, d'une page imprimée ou d'une page manuscrite.

2.3.2 Les ruptures de ligne

Quand on tape un texte sous un logiciel de *traitement de texte* comme *Word*, on pratique la frappe *au kilomètre*, c'est-à-dire qu'on ne se préoccupe pas des passages à la ligne. C'est le

[11] Ils diffèrent par exemple des fichiers qui contiennent des images, ou des fichiers qui contiennent des programmes exécutables.

logiciel qui, au cours même de la frappe, décide automatiquement d'aller à la ligne. On ne frappe sur la touche « Entrée » que lorsqu'on veut forcer le passage à la ligne, c'est-à-dire en fin de paragraphe.

Sous d'autres logiciels, qui sont aussi des *éditeurs* de fichiers textes, comme le bloc-notes de *Windows*, il est obligatoire de frapper sur la touche « Entrée » pour aller à la ligne.

À ce moment-là, le logiciel insère un (ou deux) caractère(s) particuliers, ainsi les caractères de contrôle de codes ASCII respectifs 10 et 13. Le fichier reste organisé de manière linéaire, mais les logiciels qui en gèrent l'affichage à l'écran ont les moyens de le structurer selon deux dimensions.

2.3.3 La mise en forme des documents

De nombreuses autres spécifications permettent de mettre en forme les documents afin d'obtenir des textes imprimables ayant la meilleure présentation possible.

Par exemple, on peut jouer sur la taille des caractères, avoir des caractères gras, soulignés ou en italique, obtenir des marges plus ou moins grandes, plusieurs colonnes, la justifcation ou le centrage des lignes, des notes en bas de page, la numérotation des pages, etc[12].

2.3.4 Le traitement de textes

On peut séparer en trois séries les tâches qu'accomplissent les logiciels de traitement de textes. *Word* accomplit ces différentes tâches, mais dans d'autres conceptions du traitement de textes ces tâches sont accomplies par plusieurs logiciels différents. Ainsi dans la logique de Latex, ou celle de HTML. Ces tâches sont les suivantes :

(1) L'*édition* de fichiers. Cela consiste à saisir un texte, en indiquant les mises en forme. Le texte sera codé (codage notamment de la structuration et des mises en forme du texte) selon un format qui dépend du système de traitement de textes que l'on adopte. Les marques de codage pourront être visibles à l'écran, ou pas. Les opérations de sauvegarde et de modifications des documents font partie de l'édition.

En Latex, un fichier créé de la sorte, que nous appellerons le *texte source*, aura en général une extension `tex`. Par exemple `machin.tex`. Mais, de même qu'en HTML, le logiciel d'édition n'a nul besoin d'être spécifique.

On notera qu'en Latex comme en HTML les ruptures de lignes introduites dans le texte source ne se retrouveront pas nécessairement dans le document affiché à l'écran ou imprimé.

(2) L'*affichage* à l'écran du document, en tenant compte des mises en forme.

En Latex, cette opération consiste à construire, à l'aide d'un programme, un nouveau fichier, qui aura l'extension `dvi`, par exemple `machin.dvi`. Ensuite, un autre programme permet d'afficher ce fichier à l'écran.

En HTML, l'affichage est effectué à l'aide d'un logiciel de navigation[13].

(3) L'impression sur papier du document. C'est une tâche distincte de l'affichage à l'écran, qui peut réserver des surprises, notamment quand on veut imprimer un document HTML : ce qui est

[12] On ne fera pas ici une liste exhaustive des possibilités offertes par les logiciels de traitement de texte.

[13] Cf. ci-dessus, page 11.

imprimé correspond rarement exactement à ce que le logiciel de navigation a affiché à l'écran.

Word, à l'inverse de Latex et HTML, est dans la logique d'avoir un affichage conforme à l'impression (*WYSIWYG, What You See Is What You Get*). Ce qui est plus confortable pour l'utilisateur, mais a le défaut de rendre celui-ci plus dépendant du fournisseur de logiciels. En effet, l'utilisateur ignore tout des codes utilisés pour la mise en forme, et doit avoir exactement le bon logiciel pour pouvoir lire ses fichiers et les modifier (incompatibilité même entre différentes versions de *Word*).

C'est évidemment pour des raisons commerciales que les producteurs de logiciels choisissent de telles options.

2.3.5 Documents composés de plusieurs fichiers

Un document peut être composé de plusieurs fichiers. Cela permet de scinder les documents de très grande taille en des fichiers de taille plus réduite. Ce qui est plus pratique pour la sauvegarde des fichiers, leur transmission par courrier électronique.

On peut utiliser l'option *document maître* sous *Word*, la commande `\include` en Latex, etc.

Enfin, il est possible d'inclure différents objets (images, etc.) à l'intérieur d'un document. C'est ce qui constitue une des caractéristiques les plus fondamentales de HTML.

3 HTML et l'hypertexte

3.1 Présentation

HTML (*Hyper Text Markup Language*) est un « langage » de structuration de documents, spécialement utilisé sur le Web.

Markup signifie *balise*. Une balise est désignée par son nom inséré entre deux chevrons. Par exemple `<u>` est une balise qui indique que le texte qui suit est souligné (*underlined*). Plus précisément il s'agit d'un balise *ouvrante*. Il lui correspond la balise *fermante* `</u>` qui indique que le texte à la suite n'est plus souligné.

L'objectif d'HTML est de :

- pouvoir consulter des documents composites, comprenant des textes, des graphiques, des images, des animations, des sons, et même des applications (programmes écrits dans le langage *Java*).

- d'avoir un codage indépendant des plateformes et des logiciels. L'information doit s'adapter à la taille de la fenêtre dans laquelle elle est visualisée.

La contrepartie de ceci est qu'on ne peut pas être très précis quant à la présentation des documents. Ainsi, HTML n'est pas adéquat pour l'édition d'un livre, d'un journal, ou pour l'écriture d'un article scientifique.

- de permettre la connexion entre des documents et des objets qui peuvent être physiquement éloignés les uns des autres.

Dans ce qui suit nous ne présentons qu'une partie des possibilités offertes par le langage HTML.

3.2 L'écriture des documents

3.2.1 Généralités

Dans le langage HTML, il est indifférent que les noms des balises soient écrits en majuscules ou en minuscules.

Les passages à la ligne ne sont en général pas significatifs. C'est le navigateur qui décide des passages à la ligne effectifs, selon les fenêtres dans lesquelles le texte doit être inclus.

Les balises non connues d'un navigateur seront tout simplement ignorées, sans que cela empêche de consulter le document.

3.2.2 Le codage de caractères particuliers

Un certain nombre de caractères sont obtenus de manière codée, précédés du symbole `&` et suivis d'un point-virgule. Il en est notamment ainsi des symboles "`<`" et "`>`" (utilisés pour les balises), et des lettres accentuées.

<	&lt ;	ä	ä	ô	ô	Ï	Ï
>	>	ë	ë	û	û	Ö	Ö
>	>	ï	ï	ù	ù	Ü	Ü
&	&	ö	ö	É	É	Â	Â
"	"	ü	ü	È	È	Ê	Ê
é	é	ÿ	ÿ	Ç	Ç	Î	Î
è	è	â	â	À	À	Ô	Ô
ç	ç	ê	ê	Ä	Ä	Û	Û
à	à	î	î	Ë	Ë	Ù	Ù

3.2.3 Structuration d'un document

Un document est composé d'une en-tête (*head*) et d'un corps (*body*).
Un exemple minimal de fichier HTML est :

```
<HTML>
<HEAD>
<TITLE> Sciences du langage
     </TITLE>
     </HEAD>
<BODY>
Les sciences du langage s'enseignent &agrave;
l'Universit&eacute; Paris X.
</BODY>
</HTML>
```

Le titre est connu des systèmes pour désigner de façon logique le document, mais il n'apparaît pas à l'écran. D'autres informations peuvent apparaître dans l'en-tête, en plus du titre qui, lui, est obligatoire.

3.2.4 Le format des caractères

Un certain nombre de balises permettent de donner aux caractères un format particulier. Par exemple :

`<b>`	gras
`<i>`	italique
`<u>`	souligné
`<strike>`	biffé
`<big>`	grande taille
`<small>`	petite taille
`<sub>`	indice
`<sup>`	exposant

3.2.5 La mise en forme des paragraphes

Rappelons que les ruptures de ligne dans un fichier source HTML ne sont nullement prises en compte. C'est la balise `<br>` qui insère une rupture de ligne[14]. Un paragraphe standard est placé entre `<p>` et `</p>`.

`<hr>` insère un trait de séparation, qui prend la largeur d'une ligne.

On peut centrer des paragraphes, en les plaçant entre les balises `<center>` et `</center>`.

Certains paragraphes peuvent jouer le rôle de *titre*. Il existe six niveaux de titres : `<h1>, <h2>, <h3>, <h4>, <h5>, <h6>`, qui peuvent apparaître dans n'importe quel ordre.

On peut également définir des *listes*. Plus exactement, on peut définir cinq types de listes : `<ol>, <ul>, <dir>, <menu>` sont des listes qui contiennent des `<li>` (*list items*) ; `<dl>` contient, alternativement, des termes définis `<dt>` et leur définition `<dd>`.

3.2.6 les cadres

L'écran peut être découpé en plusieurs fenêtres, grâce aux *cadres* ou *frames*. On définit un fichier source HTML, l'*index*, dans lequel la balise `<body>` est remplacée par une balise `<frameset>`.

Par exemple, le fragment de fichier HTML ci-dessous découpe l'écran en deux fenêtres, l'une à gauche prenant 25% de la largeur de l'écran, l'autre prenant 75% de cette largeur.

```
<frameset cols = "25 %, 75 %">
<frame src = "sommaire.html">
<frame src = "texte.html">
</frameset>
```

Dans chacune des fenêtres vont s'inscrire les données des fichiers spécifiés, qui sont des fichiers HTML standard.

A la place de `cols`, on peut avoir `rows`, ce qui permet de découper l'écran dans le sens de la hauteur.

Un usage fréquent des cadres est de placer dans un cadre (souvent à gauche) la table des contenus d'un document, le document proprement dit étant placé dans un autre cadre, de taille plus grande.

3.3 Les connexions

3.3.1 L'hypertexte

L'idée essentielle de l'hypertexte est de structurer des documents de manière non linéaire. Songeons à la façon dont on consulte un dictionnaire encyclopédique : la lecture d'une définition peut conduire, si on bute sur un mot que l'on ne connaît pas, ou si se trouve une référence

[14] On notera qu'à cette balise n'est pas associée une balise fermante.

à un sujet que l'on voudrait approfondir, à vouloir sauter directement en un autre point du dictionnaire.

On peut aussi vouloir regarder une figure ou une photo qui a trait à un des sujets, ou même entendre un son, déclencher une action.

Les concepteurs d'un document hypertextuel établissent des *liens* qui permettent à l'utilisateur d'effectuer très facilement des sauts à l'intérieur du document, tout simplement en cliquant sur un mot, une phrase ou une image.

Mais des liens sont établis également vers des documents externes aux documents concernés, situés sur la même machine ou sur d'autres machines connectées au réseau mondial.

L'utilisateur ne s'apercevra pas forcément de la distance entre les différents documents qu'il consulte.

3.3.2 Les liens internes

Les liens internes permettent de se déplacer à l'intérieur d'un document. Par exemple :

```
<a href="#licence">Licence mention Sciences du langage</a>
```

se traduit par l'affichage à l'écran de

Licence mention Sciences du langage

souligné, en général en bleu.

Il suffit de cliquer sur l'expression pour aller au paragraphe du document étiqueté par `maitrise`, que celui-ci soit positionné avant ou après le lien. L'étiquetage est effectué de la manière suivante :

```
<a NAME="licence"></a>
<P> La licence mention Sciences du langage se
d&eacute;roule sur trois ans.</P>
```

3.3.3 Les liens externes

Les liens externes permettent d'inclure un objet (image, son, ...) à l'intérieur d'un document. Ainsi, l'insertion d'une image peut se faire selon la syntaxe suivante :

```
<IMG SRC="photo.jpeg" height=200>
```

Le fichier `photo.jpeg` doit être dans le même répertoire que le fichier HTML où est défini le lien, sans cela il faudrait préciser le chemin d'accès.

Les liens externes permettent aussi de se déplacer vers un autre document du Web, qui peut être situé sur une toute autre machine. Par exemple :

```
<a href="http://www.education.gouv.fr/sup/univ.htm">
Enseignement sup&eacute;rieur </a>
```

va faire apparaître `Enseignement supérieur` en bleu et souligné. Si on clique sur cette expression, on chargera le document dont l'adresse est indiquée, à condition que l'adresse soit encore valable, et que la machine sur laquelle il se trouve ne connaisse aucun problème.

On peut enfin établir un lien vers un logiciel de courrier électronique. En écrivant par exemple :

```
<a href="mailto:mcori@u-paris10.fr"> Marcel Cori </a>
```

Si on clique sur `Marcel Cori` qui apparaît souligné en bleu, on est prêt à composer un message qui sera acheminé à l'adresse indiquée.

4 Les bases de données, une modélisation informatique de la réalité

4.1 Représenter des réalités organisationnelles

Un des problèmes posés par les ordinateurs est de savoir comment représenter en machine les réalités qui intéressent les êtres humains, afin d'aider les êtres humains dans leurs activités, éventuellement en se substituant à eux.

Les bases de données permettent de représenter plutôt des réalités sociales, des réalités organisationnelles, afin d'apporter une aide essentiellement aux activités de gestion, ce terme étant pris dans un sens assez large.

4.1.1 Exemples de réalités à représenter

(1) Une entreprise. Les éléments à prendre en compte sont par exemple les suivants :
- les employés, leur nom, leur prénom, leur âge, leur adresse ;
- les clients, leur nom, les commandes qu'ils ont effectuées ;
- les stocks, les articles, les quantités, les prix.

(2) Une compagnie d'aviation est une entreprise particulière, qui ne possède pas de marchandises à vendre, mais pour laquelle il faut connaître :
- les avions qu'elle a à son actif, le modèle, ainsi que le nombre de places ;
- les vols programmés, départ, arrivée, date, avion, pilote, disponibilités,...

(3) Pour une bibliothèque, il faut prendre en compte :
- les livres, l'auteur, le titre, l'année de parution, l'éditeur, savoir s'il a été prêté (et si oui à qui) ou non ;
- les emprunteurs, leur nom, leur adresse, les livres qu'ils ont empruntés.

(4) Le département de Sciences du langage de l'Université Paris X. Il faut connaître notamment :
- les enseignements assurés, leur intitulé, leur code, le nombre d'heures d'enseignement qu'ils supposent, le nombre de groupes, le diplôme auxquels ils se rattachent, les enseignants qui assurent ces enseignements, les étudiants qui y sont inscrits ;
- les étudiants, leur nom, leur prénom, leur numéro d'identification, leur adresse, le diplôme qu'ils préparent, les enseignements auxquels ils sont inscrits ;
- les enseignants, leur nom, leur prénom, leur adresse, leur statut, les enseignements qu'ils assurent.

4.1.2 Les caractéristiques de la représentation informatique d'une réalité

Le traitement par l'informatique de problèmes issus de la réalité suppose un dialogue entre informaticiens et spécialistes de domaines particuliers : gestionnaires d'entreprises, bibliothécaires, universitaires...

Seuls les spécialistes d'un domaine peuvent répondre à des questions comme :
- un avion peut-il effectuer plusieurs vols dans la même journée ?

- un auteur peut-il avoir écrit des livres chez plusieurs éditeurs différents ?
- peut-il y avoir un même livre (même titre, même auteur) avec deux dates de parution différentes ?
- un étudiant peut-il préparer deux diplômes en même temps ?
- un étudiant peut-il se réinscrire à un enseignement qu'il a déjà obtenu ?

Il est évident que les informaticiens ne doivent pas prendre des décisions qui ne les concernent pas, mais il est clair que l'informatisation d'un domaine entraîne une évolution de la vision que les spécialistes ont de leur domaine. Ils sont obligés d'énoncer très précisément leurs règles de fonctionnement, de répondre à des questions qu'ils ne s'étaient peut-être jamais posées. Il en résulte en général une certaine rationalisation de la réalité, la modification de certaines pratiques.

Par exemple, on ne peut plus décider au cas par cas (« à la tête du client ») si un étudiant a le droit de s'inscrire à deux diplômes. Ou alors il faut introduire explicitement le critère *tête du client*.

Le fonctionnement des organisations est rendu plus rigoureux, et par là même moins souple.

4.1.3 La modélisation

Les différentes réalités à décrire présentent suffisamment de similitudes pour qu'on puisse les représenter à travers un modèle général unique. Le modèle que nous allons évoquer (brièvement) ci-dessous est le modèle des relations.

Dans le cadre de ce modèle, chaque spécialiste d'un domaine donné va devoir décrire la réalité qu'il veut représenter. Il faut bien connaître cette réalité, et savoir la décrire de manière abstraite, explicite et précise.

4.2 Les systèmes de gestion de bases de données

4.2.1 Définitions

Une *base de données* est un ensemble d'informations stockées sur des mémoires périphériques qui représentent un fragment de la réalité et sont accessibles par ordinateur à travers un logiciel spécifique, le système de gestion des bases de données.

Un *système de gestion de bases de données (SGBD)* est un logiciel dont la fonction consiste en la définition et l'utilisation des bases de données. Plus précisément le SGBD doit permettre :
- de définir la structuration des données (la modélisation de la réalité) ;
- de mettre à jour les données (ajout, modification, suppression) ;
- d'accéder aux données (selon des questions, ou *requêtes*).

En général, on considère les SGBD qui traitent de très grandes quantités d'informations, situées sur différentes machines. Mais il existe des logiciels pour les ordinateurs personnels qui possèdent beaucoup des fonctionnalités des SGBD, comme *Access* de *Microsoft*.

4.2.2 Les relations

Les *relations* sont encore appelées *tables*, parce qu'on peut les décrire sous forme de tableaux. Elles permettent de représenter les différentes réalités évoquées ci-dessus.

Par exemple, pour représenter les enseignements du département de Sciences du langage, on peut définir la relation suivante, appelée `Enseignements` :

code	Intitulé	nbEcts	durée	nbGroupes
LLSDL101	Les langues	1,5	13	2
LLSDL102	Les usages	1,5	13	2
LLSDL103	Observation de faits linguistiques	3	26	6
LLSDL104	Le langage au quotidien	3	26	3
LLSDL105	Les linguistes dans le monde du travail	3	13	1
LLSDL187	Informatique	3	13	1
LLSDL196	Méthodologie du travail universitaire	3	26	4

Un deuxième tableau, que nous appellerons `groupes`, va indiquer les personnes qui assurent les différents enseignements. On remarque qu'il est nécessaire d'affecter à chaque ligne du tableau un numéro afin que ne soient pas confondus deux groupes d'un même enseignement assurés par un même enseignant.

numéro	code	enseignant
1	LLSDL101	Noyau
2	LLSDL101	Noyau
1	LLSDL102	Sitri
2	LLSDL102	Pauleau
1	LLSDL103	Kray
2	LLSDL103	Kray
3	LLSDL103	Kahane
4	LLSDL103	Kahane
5	LLSDL103	Cormier
6	LLSDL103	Tribout
1	LLSDL104	Bellonie
2	LLSDL104	de Vogüé
3	LLSDL104	Mansour
1	LLSDL105	Lacheret
1	LLSDL187	Cori
1	LLSDL196	de Vogüé
2	LLSDL196	Cormier
3	LLSDL196	Le Cunff
4	LLSDL196	Valma

Dans chaque colonne d'un tel tableau, on trouve des informations qui sont de nature semblable, ce sont des objets qui appartiennent à un même *ensemble*.

Lorsqu'on utilise un logiciel de SGBD, il faut indiquer le *type* de chaque colonne (nombre entier ou réel, suite de caractères, éventuellement la taille maximale de la suite de caractères, etc.).

Chaque ligne du tableau correspond à un élément de la relation. On peut ajouter ou supprimer des lignes sans toucher à la structure de la base de données, mais on ne peut modifier aussi simplement les colonnes.

La *clé* d'une relation est définie par le choix d'une ou plusieurs colonnes qui permettent d'identifier les lignes : il ne peut y avoir deux lignes distinctes qui aient la même valeur pour la clé.

Par exemple, dans le premier tableau, le code peut servir de clé, l'intitulé également, mais pas les autres colonnes. Dans le deuxième tableau, il faut utiliser le numéro du groupe *et* le code. On ne peut utiliser l'enseignant et le code car il y a des enseignants qui assurent deux groupes du même enseignement. On ne peut utiliser l'enseignant et le numéro car il n'est pas impossible qu'un enseignant assure deux groupes de deux enseignements différents ayant le même numéro.

4.2.3 Les requêtes

Les requêtes permettent d'extraire des informations, de manière adéquate, à partir des relations. Une requête unique peut extraire des informations provenant de plusieurs relations.

Une requête comporte trois parties :

- les relations qui servent de base à la requête ;
- les critères qui permettent de sélectionner certains éléments (et pas d'autres) dans une ou plusieurs relations ;
- les colonnes que l'on veut connaître dans les relations sélectionnées.

Les réponses fournies par les requêtes peuvent être également représentées sous forme de tableaux, mais il ne s'agit pas de tableaux conservés comme tels sur des mémoires périphériques des ordinateurs, et par conséquent pas des tableaux que l'on peut directement mettre à jour (sauf si on décide de construire une nouvelle relation à partir d'une requête).

Par exemple, une première requête peut consister à rechercher quels sont les intitulés des enseignements qui comptent plus de 20 heures. On part du tableau `enseignements`. On cherche dans la colonne `durée` les valeurs numériques supérieures à 20, mais on n'est pas obligé d'afficher ces valeurs. On peut se contenter d'afficher les intitulés, ce qui donnerait un résultat tel que :

Intitulé
`Observation de faits linguistiques`
`Le langage au quotidien`
`Méthodologie du travail universitaire`

Une autre requête peut consister à déterminer qui fait quoi, autrement dit quels sont les intitulés des enseignements assurés par les différents enseignants. Auquel cas, on part des deux relations, `enseignements` et `groupes`. Et on cherche à trouver quels sont les éléments de la première relation dont la valeur du code est égale à la valeur du code de la deuxième relation. En ce

cas, on affiche l'intitulé, qui vient de la première relation, et l'enseignant, qui vient de la deuxième relation. On trouverait le résultat suivant :

intitulé	**enseignant**
Les langues	Noyau
Les langues	Noyau
Les usages	Sitri
Les usages	Pauleau
Observation de faits linguistiques	Kray
Observation de faits linguistiques	Kray
Observation de faits linguistiques	Kahane
Observation de faits linguistiques	Kahane
Observation de faits linguistiques	Cormier
Observation de faits linguistiques	Tribout
Le langage au quotidien	Bellonie
Le langage au quotidien	de Vogüé
Le langage au quotidien	Mansour
Les linguistes dans le monde du travail	Lacheret
Informatique	Cori
Méthodologie du travail universitaire	de Vogüé
Méthodologie du travail universitaire	Cormier
Méthodologie du travail universitaire	Le Cunff
Méthodologie du travail universitaire	Valma

On peut évidemment sélectionner certaines lignes de cette réponse, selon des critères précis.

On peut également avoir des requêtes qui portent sur une colonne donnée, par exemple le nombre total de groupes qui sont assurés, ou qui portent sur une partie d'une colonne, par exemple le nombre d'heures assurées par un enseignant donné.