

Gestion de l'information multimédia

Gérard-Michel Cochard

Université Virtuelle de Tunis

2006

# Notions de HTML



## Avertissement :

Suivant le navigateur utilisé, l'affichage peut varier et certaines balises interprétées différemment, voire pas interprétées du tout !



## Structure d'un document HTML

### Qu'est-ce que HTML ?

HTML = HyperText Markup Language, langage de description de document multimédia utilisé par le Web. HTML utilise des balises ou tags pour indiquer la façon dont le document doit être affiché. Les balises sont délimitées par les signes '<' et '>'

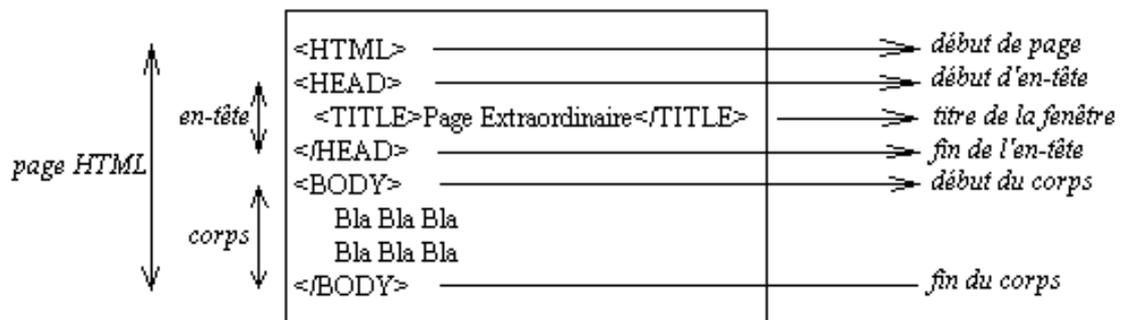
### Qu'est-ce qu'un document HTML ?

Un document HTML est un document envoyé généralement par un serveur WWW vers un poste client équipé d'un navigateur ou browser ( Netscape, Internet Explorer, Mosaic, Lynx,...) permettant de visualiser le document

Un document HTML contient des informations multimédias : textes, images fixes, sons, vidéos. Il est, en fait, un fichier spécial dont le suffixe est .htm ou .html

### Physionomie d'une page HTML

Une page HTML possède la structure de base suivante :



exemple :

la page décrite en

HTML par :

<HTML>

<HEAD>

<TITLE>

Salutation

</TITLE>

</HEAD>

<BODY>

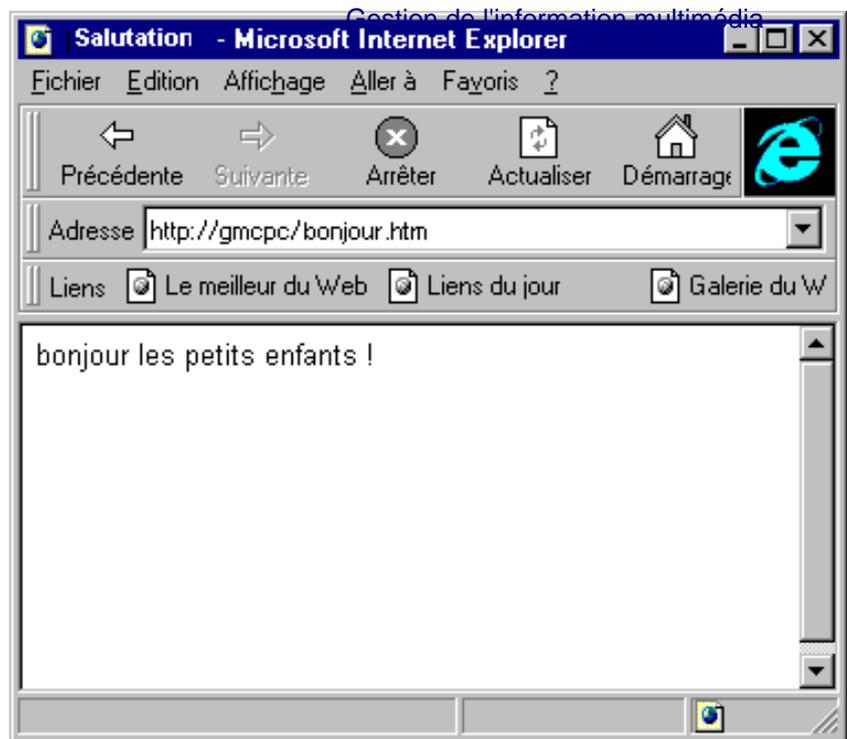
bonjour les petits

enfants !

</BODY>

</HTML>

correspond au résultat  
suivant :



## Balises usuelles

### Styles physiques de texte

<B>bonjour</B>	gras	<b>bonjour</b>
<I>bonjour</I>	italique	<i>bonjour</i>
<U>bonjour</U>	souligné	<u>bonjour</u>
<S>bonjour</S>	barré	<del>bonjour</del>
bon<SUB>jour</SUB>	indice	bon <sub>jour</sub>
bon<SUP>jour</SUP>	exposant	bon <sup>jour</sup>

### Styles logiques de texte

<H1>bonjour</H1>	titre taille 1	<b>bonjour</b>
<H2>bonjour</H2>	titre taille 2	<b>bonjour</b>
<H3>bonjour</H3>	titre taille 3	<b>bonjour</b>
<H4>bonjour</H4>	titre taille 4	<b>bonjour</b>
<H5>bonjour</H5>	titre taille 5	<b>bonjour</b>
<H6>bonjour</H6>	titre taille 6	<b>bonjour</b>
bonjour<EM>bonsoir</EM>	mise en valeur	<i>bonjourbonsoir</i>
 	saut de ligne	
<P>	paragraphe	

## Alignement de texte

Il est défini par un attribut `ALIGN = valeur` où valeur peut être `RIGHT`, `CENTER`, `LEFT`. L'attribut `ALIGN` se place dans un tag `<P>` ou `<Hx>` ou `<DIV>` ou `<HR>`

exemple : `<P ALIGN=CENTER>`

## Taille des caractères

Elle est définie par la balise `BASEFONT` :

`<BASEFONT SIZE = x>` où `x = 1, 2, 3, 4, 5, 6, 7`

## Tracé d'une ligne horizontale

Il est défini par la simple balise `<HR>` qui peut posséder les attributs suivants :

`WIDTH` : longueur de la ligne en %  
`SIZE` : épaisseur en pixels  
`ALIGN = x`

## Couleur du texte

Elle est définie par l'attribut `COLOR` de la balise `<FONT>` ; les valeurs de `COLOR` sont : `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white`, `yellow`

## Couleur du fond

Elle est définie par l'attribut `BGCOLOR` dans le tag `<BODY>` pour la couleur de fond (qui est donc constante dans un même document).

## Couleur des liens

Les attributs pour les liens dans le tag `<BODY>` sont

`LINK` : liens non visités  
`VLINK` : liens visités  
`ALINK` : liens actifs

## Un exemple

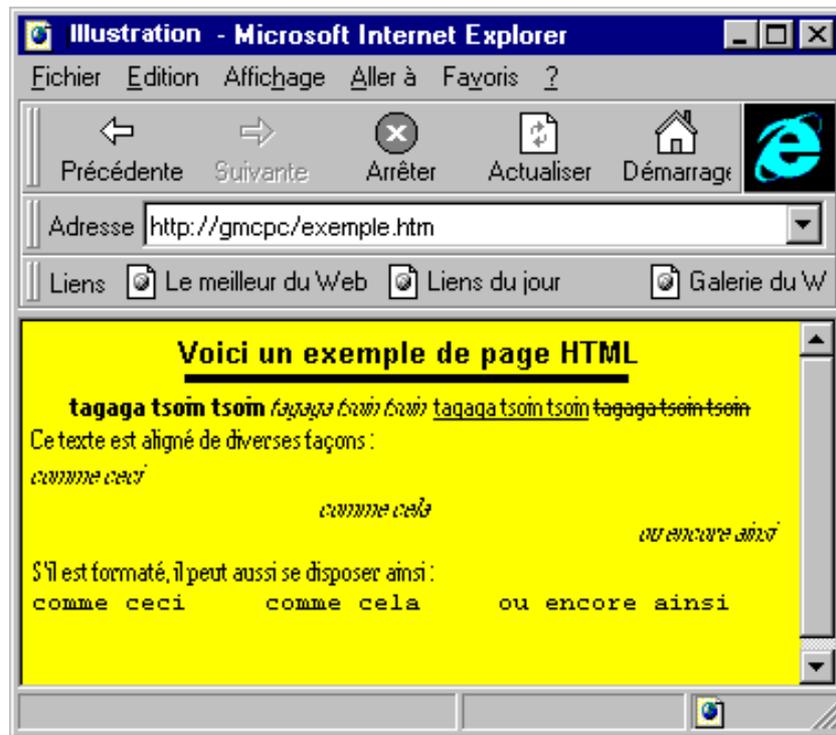
```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
<HEAD>
<TITLE>Voici un exemple de page HTML</TITLE>
</HEAD>
<BODY BGCOLOR="yellow">
<H2 ALIGN="center">Voici un exemple de page HTML</H2>
<HR SIZE="10" WIDTH="50%">
<P ALIGN="center"><B>tagada tsoin tsoin</B>
<I>tagada tsoin tsoin</I>
<U>tagada tsoin tsoin</U>
```

```

<S></S></P>
<P>Ce texte est aligné de diverses façons :</P>
<P><EM>comme ceci </EM></P>
<P ALIGN="center"><EM>comme cela </EM></P>
<P ALIGN="right"><EM>ou encore ainsi</EM></P>
<P ALIGN="left">S'il est formaté, il peut aussi se disposer ainsi :</
P>
<PRE>comme ceci    comme cela ou encore ainsi</PRE>
</BODY>
</HTML>

```

donne le résultat suivant :



## Listes

### Listes numérotées

Ce sont des listes d'informations présentées suivant un ordre défini par un numéro

balises pour la liste numérotée : <OL>.....</OL>

balises pour le titre de la liste : <LH>.....</LH>

balises pour les items de la liste : <LI>

exemple :

```
<OL>Université Virtuelle de Tunis
```

Gestion de l'information multimédia

```
<B><LH>Opérations à
```

```
effectuer :
```

```
</LH></B>
```

```
<LI>Prendre son couteau
```

```
<LI>Prendre sa fourchette
```

```
<LI>Couper un morceau de
```

```
steack
```

```
<LI>Le manger
```

```
<LI>Boire un peu de vin
```

```
</OL>
```

correspond à

**Opérations à effectuer :**

1 - Prendre son couteau

2 - Prendre sa fourchette

3 - Couper un morceau de steack

4 - Le manger

5 - Boire un peu de vin

**Listes non numérotées**

Ce sont des listes analogues aux précédentes, mais où des puces remplacent les numéros.

balises à utiliser : <UL>.....</UL>

attribut de définition de puce : TYPE = SQUARE|CIRCLE|DISC

exemple :

```
<ul>
```

```
<b><LH>Opérations à
```

```
effectuer : </LH></b>
```

```
<li TYPE=SQUARE> Prendre son
```

```
couteau </li>
```

```
<li> Prendre sa fourchette </
```

```
li>
```

```
<li TYPE=SQUARE>
```

```
Couper un morceau de steack
```

```
</li>
```

```
<li TYPE=SQUARE>Le manger </
```

```
li>
```

```
<li TYPE=CIRCLE>Boire un peu
```

```
de vin </li>
```

```
</ul>
```

correspond à

**Opérations à effectuer :**

■ Prendre son couteau

■ Prendre sa fourchette

■ Couper un morceau de steack

■ Le manger

○ Boire un peu de vin

**Listes de définition**

Ce sont des listes permettant de faire des glossaires

balises pour une liste de définition : <DL>.....</DL>

balises pour le sujet : <DT>.....</DT>

balises pour la définition : <DD>.....</DD>

exemple :

```

<B>Université Virtuelle de Tunis</B>
<DL>
<DT>alpha</DT>
<DD>première lettre de
l'alphabet grec</DD>
<DT>beta</DT>
<DD>deuxième lettre de
l'alphabet grec</DD>
<DT>gamma</DT>
<DD>troisième lettre de
l'alphabet grec</DD>
</DL>

```

## Dictionnaire

alpha  
première lettre de l'alphabet grec

beta  
deuxième lettre de l'alphabet grec

gamma  
troisième lettre de l'alphabet gre

correspond à



## Tableaux

### Définition d'un tableau

Un tableau est constitué de lignes et de colonnes. Les balises qui délimitent un tableau sont :

```

<TABLE>.....</TABLE>
attribut : BORDER = 1|2|3.....
attribut : ALIGN = BLEEDLEFT|BLEEDRIGHT|CENTER|LEFT|RIGHT|JUSTIFY
balises pour le titre : <CAPTION>.....</CAPTION> avec l'attribut : ALIGN = TOP|BOTTOM
balises pour chaque ligne : <TR>.....</TR> avec l'attribut : ALIGN = CHAR|CENTER|DECIMAL|LEFT|
RIGHT|JUSTIFY
balises pour les cases de titres de lignes ou de colonnes : <TH>.....</TH>
avec l'attribut : ALIGN = CHAR|CENTER|DECIMAL|LEFT|RIGHT|JUSTIFY
et l'attribut : VALIGN = TOP|MIDDLE|BOTTOM|BASELINE
balises pour les données : <TD>.....</TD> avec l'attribut : ALIGN = CHAR|CENTER|DECIMAL|LEFT|
RIGHT|JUSTIFY
et l'attribut : VALIGN = TOP|MIDDLE|BOTTOM|BASELINE

```

exemple :

```

<TABLE BORDER =2 bgcolor="silver">
<CAPTION ALIGN = BOTTOM>
<B><I>Internet casse les prix du téléphone</I></B></
CAPTION>
<CAPTION ALIGN = TOP><U><B>L'actualité</B></U></CAPTION>
<P>
<TR ALIGN=CENTER><TH>Destination</TH>
<TH>1 min par France Telecom</TH>
<TH>1 min par Internet</TH>
<TH>Economie réalisée</TH></TR>
<TR ALIGN=CENTER><TH>Paris-New York</TH>
<TD>2,25F</TD><TD>0,85F</TD><TD>62%</TD></TR>
<TR ALIGN=CENTER><TH>Paris-Tokyo</TH>
<TD>5,90F</TD><TD>1,80F</TD><TD>69%</TD></TR>
</TABLE>

```

correspond au tableau suivant :

### L'actualité

*Internet casse les prix du téléphone*

Destination	1 min par France Telecom	1 min par Internet	Economie réalisée
Paris-New York	2,25F	0,85F	62%
Paris-Tokyo	5,90F	1,80F	69%

## Cellspanning, Cellspacing, Cellpadding

Cellspanning effectue la fusion de plusieurs cellules et utilise les attributs : COLSPAN et ROWSPAN

Cellspacing gère l'espacement des cellules et utilise l'attribut : CELLSPACING = 0|1|2|3.....

Cellpadding gère l'espacement du texte d'une cellule par rapport au bord et utilise l'attribut : CELLPADDING = 0|1|2|3.....

exemple 1 :

```
<TABLE BORDER=1 ALIGN=CENTER>
<CAPTION ALIGN = TOP>Cellspanning</CAPTION>
<TR ALIGN = center >
<TH>....</TH><TH>colonne 1</TH><TH>colonne 2</TH>
<TH>colonne 3</TH></TR>
<TR ALIGN = Center ><TH >ligne 1</TH>
<TD COLSPAN = 2>11 et 12</TD><TD>13</TD></TR>
<TR ALIGN = Center ><TH>ligne 2</TH><TD>21</TD>
<TD COLSPAN = 2 ROWSPAN=2>22 et 23 et 32 et 33</TD>
</TR>
<TR ALIGN = Center ><TH>ligne 3</TH><TD>31</TD></TR>
</TABLE>
```

correspond à :

Cellspanning

....	colonne 1	colonne 2	colonne 3
<b>ligne 1</b>	11 et 12	13	
<b>ligne 2</b>	21	22 et 23 et 32 et 33	
<b>ligne 3</b>	31		

exemple 2 : Cellspanning, Cellpadding

Le code HTML suivant

```
<H4><I>Cellspacing et Cellpadding</I></H4>
<P><TABLE ALIGN=center BORDER = 1 CELLSPACING = 10 CELLPADDING = 10 >
<TR >
<TD >Cellspacing<BR>=10
</TD>
<TD>Cellpadding<BR>=10
</TD>
<TD>
Border<BR>=1
</TD>
</TR>
</TABLE>
<TABLE ALIGN =center BORDER = 2 CELLSPACING = 10 CELLPADDING =0>
<TR>
<TD>
Cellspacing<BR>=10
```

```
</TD>Université Virtuelle de Tunis
```

```
<TD>
```

```
Cellpadding<BR>=0
```

```
</TD>
```

```
<TD>
```

```
Border<BR>=2
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
<BR><BR><BR><BR>
```

```
<TABLE ALIGN=center BORDER = 3 CELLSPACING = 0 CELLPADDING = 10>
```

```
<TR>
```

```
<TD >
```

```
Cellspacing<BR>=0
```

```
</TD>
```

```
<TD>
```

```
Cellpadding<BR>=10
```

```
</TD>
```

```
<TD>
```

```
Border<BR>=3
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE ALIGN=center BORDER = 4 CELLSPACING = 0 CELLPADDING = 0>
```

```
<TR>
```

```
<TD>
```

```
Cellspacing<BR>=0
```

```
</TD>
```

```
<TD>
```

```
Cellpadding<BR>=0
```

```
</TD>
```

```
<TD>
```

```
Border<BR>=4
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

correspond au résultat :

### ***Cellspacing et Cellpadding***

Cellspacing =10	Cellpadding =10	Border =1
Cellspacing =10	Cellpadding =0	Border =2
Cellspacing =0	Cellpadding =10	Border =3
Cellspacing =0	Cellpadding =0	Border =4



## Images

### Formats graphiques

2 formats graphiques seulement sont acceptés par l'ensemble des navigateurs :

- GIF : Graphics Interchange Format d'origine CompuServe  
256 couleurs max  
image compressée sans perte
- JPEG : Joint Photographic Experts Group  
16,7 millions de couleurs max  
image compressée avec perte

D'autres formats nécessitent des extensions des navigateurs appelés plugs-in : DIB, TIFF, PCX, WMF, BMP, TGA.

### Images progressives, images transparentes

- format GIF entrelacé : affichage de 1 ligne sur x : affichage progressif de l'image : ergonomie accentuée.
- avec GIF on peut définir la couleur de l'image comme transparente d'où la possibilité d'effets spéciaux.

### La balise IMG

L'insertion d'une image est annoncée par le tag <IMG> avec un certain nombre d'attributs :

- attribut SRC : indique la source de l'image. ex : SRC = C:\truc\machin\image4.gif
- attribut WIDTH : indique la largeur de l'image
- attribut HEIGHT : indique la hauteur de l'image. ex : WIDTH = 20 HEIGHT = 50
- attribut ALIGN : valeurs TOP (en haut), MIDDLE (au milieu de la ligne), BOTTOM (en bas).
- attribut ALT : texte alternatif pour les navigateurs qui ne peuvent afficher l'image
- attribut BORDER : largeur de la bordure de l'image
- attributs VSPACE et HSPACE : création d'espaces vides autour de l'image

exemple :

### Quelques images

```
<H3 align="center"><B>Quelques
images</B></H3>
```

```
<IMG SRC= "ampoule.gif"
BORDER="1" HSPACE="20"
VSPACE="20" WIDTH="70"
HEIGHT="66">
```

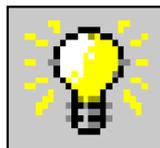
```
< IMG SRC ="ampoule.gif" BORDER
="1 " VSPACE= "20" WIDTH ="140"
HEIGHT ="66">
```

```
<BR>
```

```
< IMG SRC ="ampoule.gif" BORDER
="1" HSPACE="20" WI DTH="70"
HEIGHT ="132">
```

```
< IMG SRC ="ampoule.gif" BORDER
="1" WIDTH ="140" HEI GHT ="132">
```

correspond à

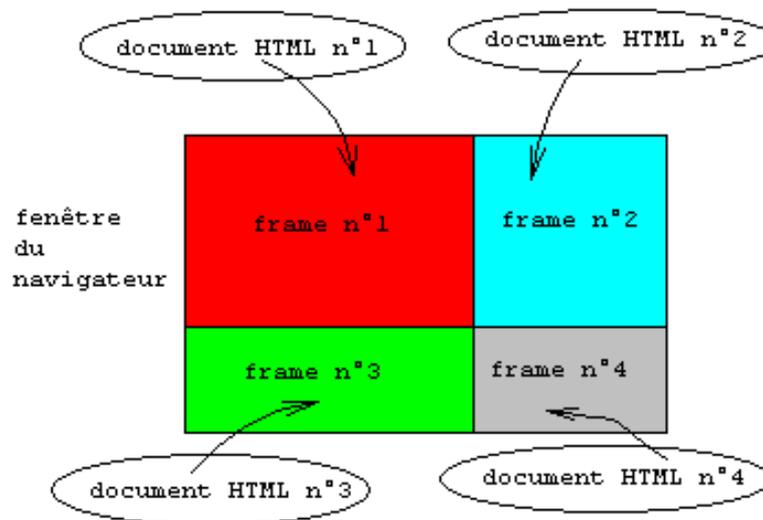




## Multifenêtrage

### La technique des frames

Les frames (en français cadres) permettent d'afficher plusieurs documents HTML à l'intérieur de visionneuses prévues dans la fenêtre d'un navigateur :



### Les balises FRAMESET et FRAME

définition du nombre et de la taille des frames : `<FRAMESET>.....</FRAMESET>`

attributs COLS et ROWS avec valeur en % ou en pixels

ex : `<FRAMESET COLS=«100,25%,*»` crée 3 frames dans le sens de la largeur, la première de 100 pixels, la deuxième du quart de la largeur totale, la troisième de ce qui reste.

définition du document HTML à afficher dans la frame : `<FRAME>`

attributs : NAME qui indique le nom de la frame

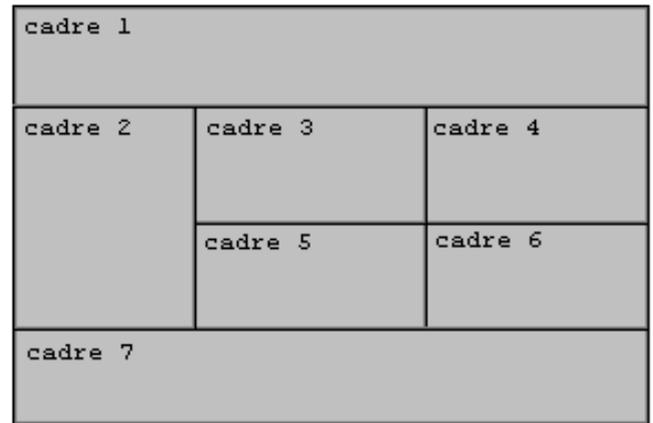
SRC qui indique l'adresse du document HTML à afficher

exemple :

```

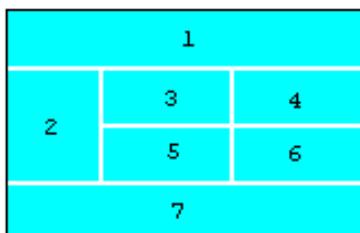
<html>
<head>
<title>essai8</title>
</head>
<frameset rows="25%,50%,25%">
  <frame src="cadre1.htm">
  <frameset cols="25%,75%">
    <frame src="cadre2.htm">
    <frameset rows="50%,50%" cols="50%,50%">
      <frame src="cadre3.htm">
      <frame src="cadre4.htm">
      <frame src="cadre5.htm">
      <frame src="cadre6.htm">
    </frameset>
  </frameset>
  <frame src="cadre7.htm">
</frameset>
</html>

```



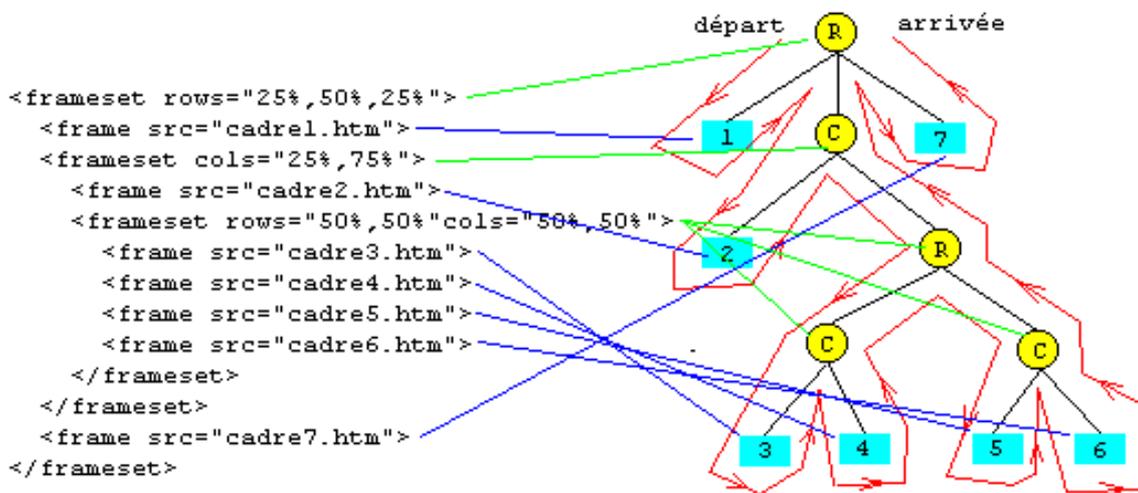
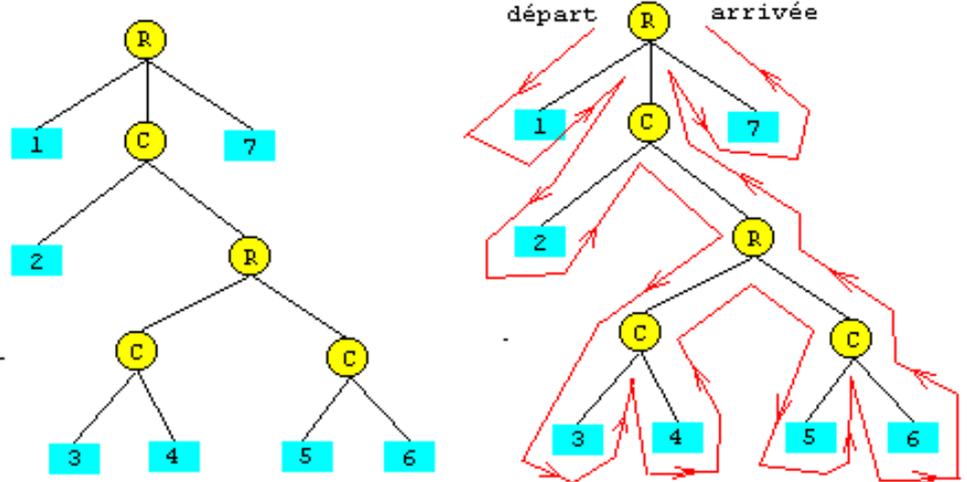
## Méthode pour imbriquer des frames

La numérotation des frames s'opère de bas en haut et de gauche à droite : cette numérotation résulte du PTT (Pre-order Tree Traversal), sens de parcours d'un arbre permettant de numérotter les feuilles.



représentation  
par un arbre

R = subdivision en lignes  
C = subdivision en colonnes



## Autres balises et attributs

- dans la balise <FRAME> :

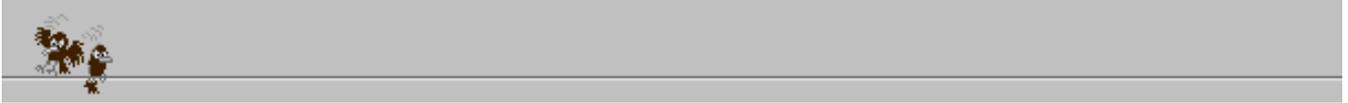
attribut SCROLLING =YES|NO|AUTO ajout/non ajout d'ascenseurs

attribut NORESIZE blocage du redimensionnement des frames par l'utilisateur

attributs MARGINHEIGHT et MARGINWIDTH définition d'espacements (en pixels) par rapport au bord de la frame

- dans les balises <FRAME> ou <FRAMESET> :

attribut FRAMEBORDER=YES|NO visibilité/non visibilité des bords de la frame



## Hyperliens

### Hypertexte, hypermedia

Avec HTML on peut créer des liens entre documents WWW ou objets de documents WWW. Ces liens sont appelés Hyperliens. L'hypertexte en est un exemple; c'est un texte qui est sensible au clic de la souris et qui provoque une action sous ce clic (en général, le chargement d'un autre document WWW). Comme on peut aussi cliquer sur une image pour provoquer une action, on peut alors parler d'hypermédia...

### Ancres

Une ancre est l'indication d'une cible par une balise <A>. La cible peut être :

- a) une étiquette dans un document local ou distant  
définition de l'étiquette par un nom: <A NAME=«truc»>.....</A>
- b) un document local ou distant  
le document est défini par son adresse URL, par exemple <http://www.u-picardie.fr/>

L'hyperlien est défini dans une balise <A> par l'attribut HREF :

<A HREF=«#truc»>.Aller à truc </A> : lien local vers une étiquette «truc»

<A HREF=«http://www.u-picardie.fr»>.univ. de picardie</A> : lien vers un serveur WWW (page d'accueil)

<A HREF=«http://www.nando.net/toys/cyrano.html»>Ecrivez des lettres d'amour avec Internet</A> : lien vers un document html d'un serveur WWW

<A HREF=«http://www.cenaath.cena.dgac.fr/themes/cuisine/plat/#veau»>Recettes de cuisine</A> : lien vers une étiquette d'un document d'un serveur WWW

exemple :

```
<html>
<head>
<title>exemples de liens</title>
</head>
<body bgcolor="#FFFFFF">
Ceci est un exemple d'utilisation de liens<BR>
<a name="Table">Sommaire</a>
<ul>
<li><a href="#chapitre1">Chapitre 1</a> </li>
<li><a href="#chapitre2">Chapitre 2</a> </li>
<li><a href="#chapitre3">Chapitre 3</a> </li>
<li><a href="#conclusion">Conclusion</a></li>
</ul>
```

```

<p>blablabla</p>
<p>blablablabla</p>
<h3><a name="chapitre1">Chapitre 1</a></h3>
<p>blablabla</p>
<p>blablabla</p>
<p>blablabla</p>
<p><a href="#Table">Retour au sommaire</a><BR>
<h3><a name="chapitre2">Chapitre 2</a> </h3>
<p>blablabla</p>
<p>blablabla</p>
<p>blablabla</p>
<p><a href="#Table">Retour au sommaire</a>
<BR>
<h3><a name="chapitre3">Chapitre 3</a> </h3>
<p>blablabla</p>
<p>blablabla</p>
<p>blablabla</p>
<p><a href="#Table">Retour au sommaire</a><BR>
<h3><a name="conclusion">Conclusion</a> </h3>
<p>blablabla</p>
<p>blablabla</p>
<p>blablabla</p>
<p><a href="#Table">Retour au sommaire</a><BR>
<p><a href="http://www.u-picardie.fr/">Cliquez ici pour visiter notre site</a>
<p><a href="mailto:cochard@u-picardie.fr">Ecrivez-moi</a>
</body>
</html>

```

## Images réactives

On peut remplacer un «hypermot» par une «hyperimage» ou image réactive. Une image réactive se définit par la balise <MAP> qui possède l'attribut NAME nom de l'image réactive et par la balise <AREA> qui possède les attributs : SHAPE forme géométrique de l'image réactive (circle, rect, polygon), COORDS coordonnées des points de contrôle de l'image réactive, HREF adresse de destination.

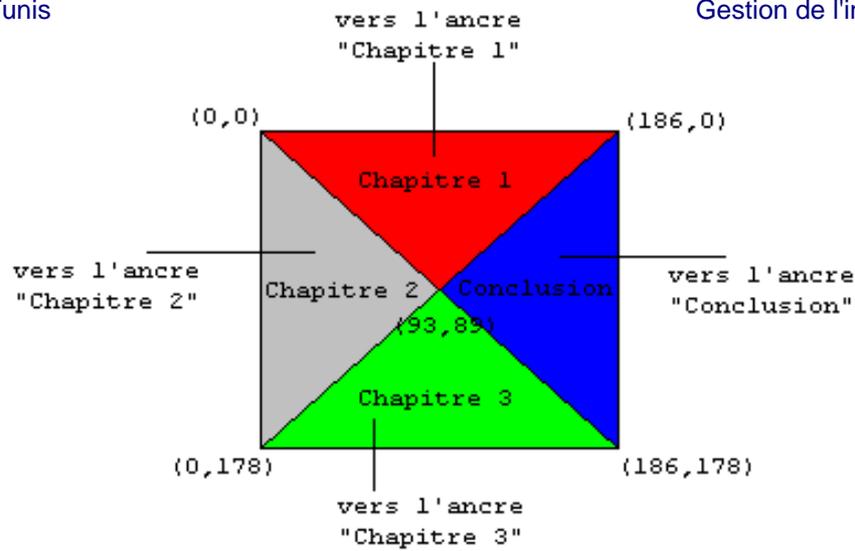
exemple : le code ci-dessous

```

<MAP NAME=" ImageReac0">
<AREA SHAPE="POLYGON" COORDS="93, 89, 0, 0, 186, 0" HREF="#chapitre1">
<AREA SHAPE="POLYGON" COORDS="0, 0, 0, 178, 93, 89" HREF="#chapitre2">
<AREA SHAPE="POLYGON" COORDS="0, 178, 186, 178, 93, 89" HREF="#chapitre3">
<AREA SHAPE="POLYGON" COORDS="186, 0, 93, 89, 186, 178" HREF="#conclusion">
</MAP>


```

correspond à la figure réactive suivante (4 zones sensibles)



## Formulaires

### Principe général

Les formulaires sont utilisés pour échanger des données entre le poste client et le serveur. Cet échange de données fait appel à des programmes spéciaux s'exécutant sur le serveur. Dans la méthode standard, ils sont appelés scripts CGI (CGI = Common Gateway Interface). Il existe d'autres méthodes, mais pour expliquer les formulaires, on se bornera à CGI.

1) L'utilisateur remplit le formulaire et le valide ce qui a pour effet d'envoyer des données vers le script CGI situé sur le serveur



2) Le script CGI sur le serveur accepte les données comme paramètres et effectue un traitement produisant éventuellement des résultats



3) Les résultats sont transmis (éventuellement) au poste client. Ils sont affichés par le navigateur



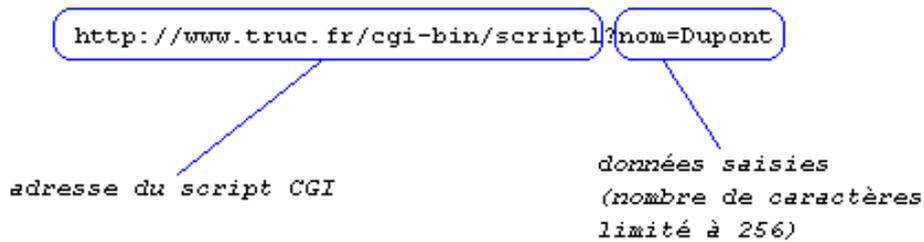
## Envoi de paramètres

Un formulaire est délimité par les balises `<FORM>` et `</FORM>`. Les champs de saisie se trouvent entre ces deux balises. Les données saisies sont transmises selon deux méthodes indiquées par l'attribut `METHOD` au script CGI (sur le serveur)

### a) METHOD = GET

envoi des données saisies (paramètres) à l'adresse du script CGI :

exemple :



### b) METHOD = POST

permet l'envoi de listes de données plus longues (les paramètres sont envoyés indépendamment de l'adresse URL du script).

## Structure d'un formulaire

L'adresse du script CGI est défini par l'attribut `ACTION` ; exemple : `ACTION=<http://www.truc.fr/cgi-bin/script1>`. Les zones de saisie sont définies par la balise `<INPUT>` qui possède les attributs suivants :

`NAME` : nom du champ de saisie ;

`TYPE` : type de champ ;

text : texte ;

password : mot de passe

checkbox : boîte à cocher

radio : boîte d'option

`SIZE` : taille du champ

`MAXLENGTH` : nombre maximum de caractères à saisir

Le déclenchement de l'envoi des paramètres peut s'effectuer avec un bouton «Envoyer» :

```
<INPUT TYPE=SUBMIT VALUE="Envoyer" >
```

La remise à leur valeur initiale des paramètres peut également s'effectuer avec un bouton «Reset»

```
<INPUT TYPE=RESET VALUE="Rétablir" >
```

exemple :

```
<FORM METHOD=GET ACTION=«http://www.truc.fr/cgi-bin/inscription»>
```

Merci de bien vouloir indiquer votre nom :

```
<INPUT NAME=«Nom» TYPE=Text SIZE=32 MAXLENGTH=32>
```

```
<P><INPUT TYPE=SUBMIT VALUE=«Envoyer»>
```

```
<P><INPUT TYPE=RESET VALUE=«Rétablir»>
```

```
</FORM>
```

correspond au formulaire :

Merci de bien vouloir indiquer votre nom :

## Différents types de formulaires

- champ de saisie sur une ligne

```
<INPUT NAME=«Truc» TYPE=Text SIZE=20 MAXLENGTH=12>
```

- saisie d'un mot de passe

```
<INPUT NAME=«Truc» TYPE=Password SIZE=20 MAXLENGTH=12>
```

- saisie sur plusieurs lignes

```
<TEXTAREA NAME=«Baratin» ROWS="4" COLS="20">
Ecrire ici</TEXTAREA>
```

- cases à cocher

```
<INPUT NAME=«pain» TYPE=CHECKBOX> pain
<INPUT NAME=«vin» TYPE=CHECKBOX CHECKED> vin
<INPUT NAME=«potage» TYPE=CHECKBOX> potage
<INPUT NAME=«salade» TYPE=CHECKBOX> salade
```

pain  
vin  
potage  
salade

- cases à option ou boutons radio (sélection d'un seul élément)

brûlant

chaud

tiède

froid

```
<INPUT NAME=«X» TYPE=RADIO VALUE=«4»>brûlant
<INPUT NAME=«X» TYPE=RADIO VALUE=«3»>chaud
<INPUT NAME=«X» TYPE=RADIO VALUE=«2»>tiède
<INPUT NAME=«X» TYPE=RADIO CHECKED VALUE=«1»>froid
```

- menu déroulant (sélection d'un item ou de plusieurs items)

```
<SELECT NAME=«légume» SIZE=1 MULTIPLE>
<OPTION VALUE=«Salsifi» SELECTED>Salsifi
<OPTION VALUE=«Pomme de terre»>Pomme de
terre
<OPTION VALUE=«Carotte»>Carotte
</SELECT>
```

SI ZE indiquez le nombre d'options visibles

MULTIPLE indique une sélection multiple ; pour une sélection unique, cet attribut est omis.



## Bibliographie

J.C. HANKE

P. CHALEAT, D. CHARNAY

HTML 4 XML

Programmation HTML et JavaScript

Micro Application

Eyrolles

# Introduction à XML



## Quelques définitions

### SGML

SGML (Standard Generalized Markup Language) est un métalangage dont l'objectif est la description de langages utilisant des balises (par exemple HTML est un langage Markup). SGML a été inventé pour afficher un document quelle que soit la plate-forme de lecture. SGML prévoit qu'un document possède une structure définie par un fichier spécifique appelé DTD (Définition de Type de Document ou Document Type Definition). Dans un langage décrit par SGML, il faut donc produire

- 1) le document (dont la structure est décrite par des balises)
- 2) la DTD (qui définit les balises)

ce qui peut représenter un mode opératoire assez lourd.

Pour plus d'informations consulter : <http://www.oasis-open.org/cover/general.html>

### HTML

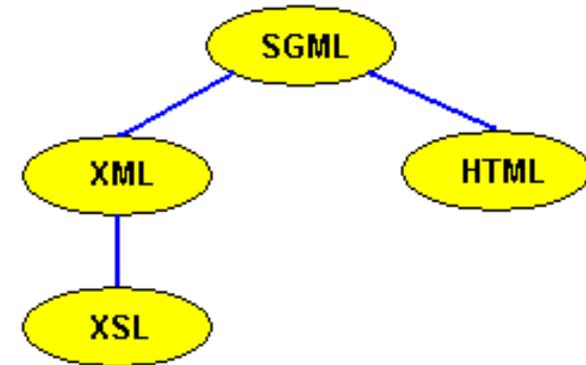
HTML, inventé par Tim Berners Lee, est basé sur une DTD particulière, comprise par les navigateurs et qu'il n'est pas besoin de transmettre. HTML est un langage de présentation de document multimédia ayant une double fonctionnalité :

- définition de la structure du document (par exemple balise <p>)
- mise en page du document (par exemple balise <font>)

HTML utilise un certain nombre de balises prédéfinies, bien connues des développeurs sur Internet.

## XML

XML (Extensible Markup Language) est une simplification de SGML. C'est donc toujours un métalangage et il peut décrire HTML par exemple. Son objectif essentiel, comme celui de SGML, est la description de la structure d'un document.



**la filiation des "Markup Languages"**

Un document XML est un document créé avec les spécifications de XML. Il possède des balises qui sont décrites dans une DTD. Il n'y a donc plus, comme dans HTML, des balises prédéfinies : XML crée des balises à volonté.

La mise en page n'est pas du ressort de XML ; elle est définie par des fichiers annexes : CSS ou XSL.

Documentation : <http://www.w3.org/XML/> ; [www.w3schools.com/xml/default.asp](http://www.w3schools.com/xml/default.asp)

### Document valide

Il s'agit d'un document utilisant des balises définies dans une DTD.

### Document bien formé

Il s'agit d'un document HTML compatible avec XML, c'est à dire respectant un certain nombre de règles dont les principales sont :

- l'imbrication des balises : le début et la fin d'une balise ne peut être que dans une zone limitée par le début et la fin d'une autre balise.

mauvais exemple (mal formé)

```
<body><p>.....  
.....  
</body></p>
```

bon exemple (bien formé)

```
<body><p>.....  
.....  
</p></body>
```

- les noms de balise doivent respecter la casse

<coucou>.....</Coucou> ne respecte pas la casse car la balise de début et la balise de fin n'ont pas la même écriture.

En règle générale, depuis quelque temps, toutes les balises de XML doivent être formulées en minuscules.

- toutes les balises doivent être fermées.

Les balises qui n'ont pas de "fin" doivent être terminées par un "/". Ainsi la balise de HTML qui définit l'affichage d'une image doit être formulée comme suit :

```
<img ..... />
```

- tous les attributs doivent être définis entre guillemets

A noter qu'un document bien formé peut ne pas être valide s'il n'y a pas de DTD associée.

## XHTML

Ce sigle désigne du HTML respectant les règles relatives à un document bien formé.

## MathML, SMIL et Cie

XML permet de définir des langages Markup spécialisés à certains usages :

- MathML (Mathematical Markup Language) : langage adapté à la description des formules de mathématiques
- SMIL (Synchronized Multimedia Integration Language) : langage permettant l'intégration de données multimédias (images animées, sons,...).
- CML ( Chemical Markup Language) : langage de description de molécules
- VML (Vectorial Markup Language) : langage pour le dessin vectoriel
- MusicML : définition de partitions musicales
- VoxML : application pour le langage parlé (<http://www.voxml.com> )
- HRML (Human Resources Markup Language) : application de présentation d'offres d'emploi
- WML (WAP Markup Language) : langage destiné aux mobiles pour l'accès Internet



## Balises, éléments, entités

Un exemple de document bien formé est donné ci-dessous :

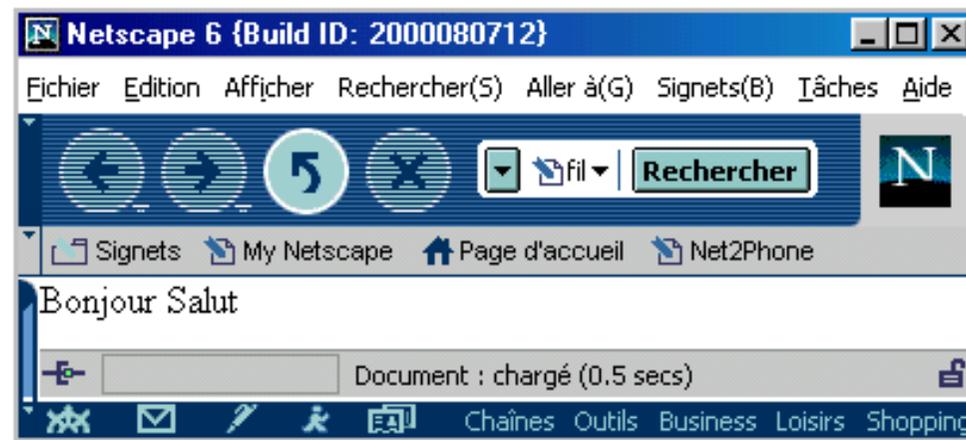
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<document>
<question>Bonjour</question>
<reponse>Salut</reponse>
</document>
```

Ce fichier constitué avec n'importe quel éditeur de texte est enregistré sous le nom de salutations.xml. La première ligne (prologue) indique qu'il s'agit d'un document XML dont la version est 1.0 et dont le jeu de caractères est Latin 1.

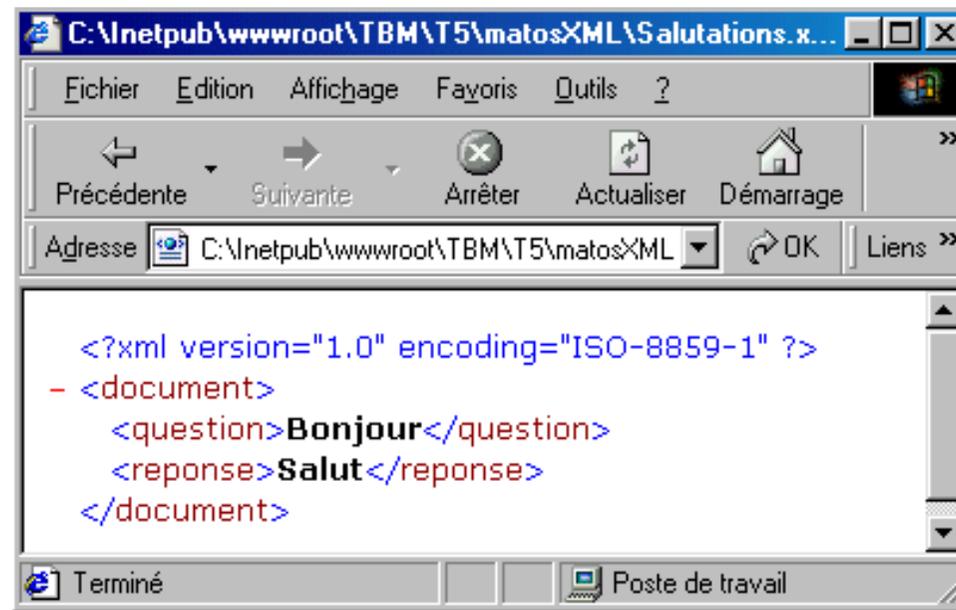
<document> est la première balise rencontrée. Elle correspond à l'élément "racine". Un document XML a toujours un élément racine et un seul. Les autres balises, utilisées dans le document, sont <questions> et <reponse>.

Si on affiche ce fichier (auquel, pour l'instant, on n'a pas affecté de DTD) dans un navigateur, on obtiendra des résultats très différents :

- avec Netscape 6, on obtient bien ce qu'on attend (la mise en page n'étant pas notre préoccupation pour l'instant) :



- avec Internet Explorer5, on obtient une présentation structurée :

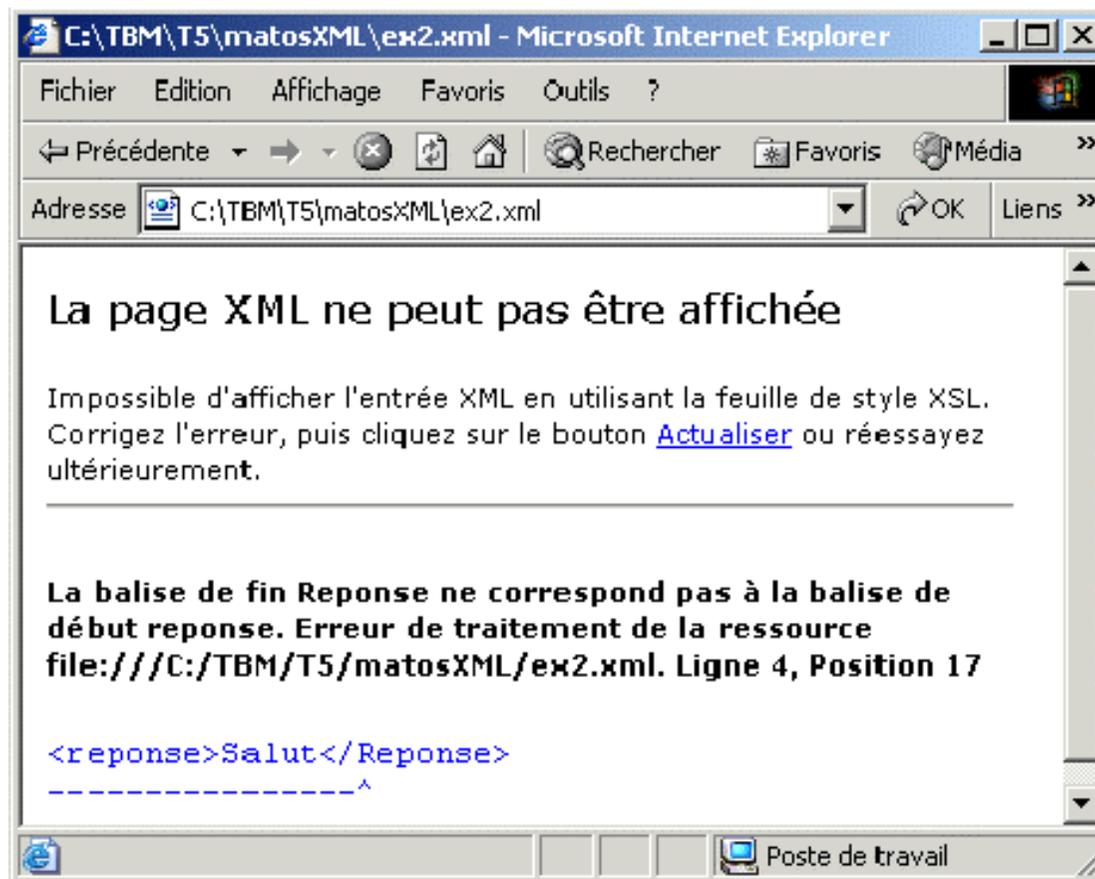


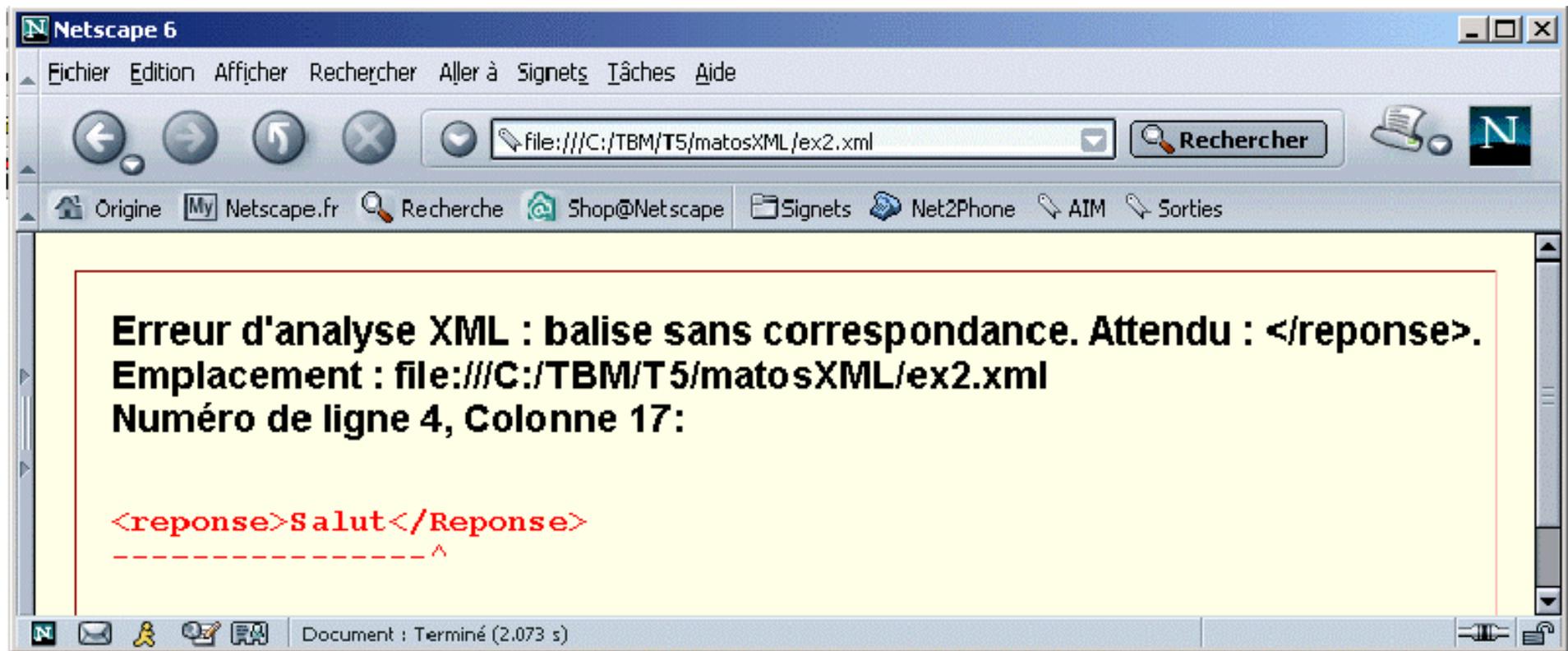
Nous voyons donc que IE5 permet de visualiser la structure d'un document XML.

Imaginons maintenant un document incorrect comme le suivant (pourquoi n'est-il pas bien formé ?) :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<document>
<question>Bonjour</question>
<reponse>Salut</Reponse>
</document>
```

L'affichage dans un navigateur permet la détection de l'erreur :





On constate qu'un code XML est composé exclusivement d'**éléments**, c'est à dire de blocs constitué d'une balise d'ouverture, d'un contenu et d'une balise de fermeture : `<tagada>.....</tagada>`. Les éléments peuvent être

- simples :

```
<tagada>tsoinstsoin</tagada>
```

- composés :

```
<tagada>il y a des éléments fils
  <tsoin>en voici un</tsoin>
  <tsoin>en voici un autre</tsoin>
</tagada>
```

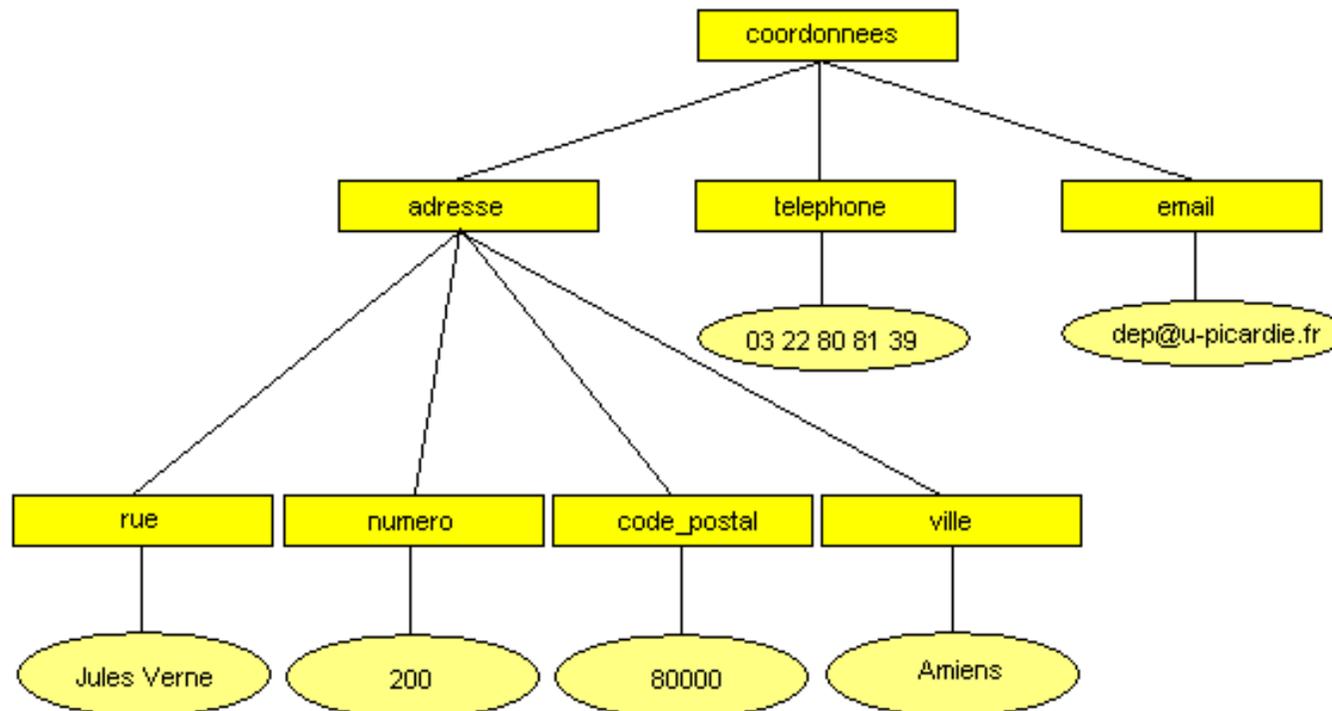
- vides :

```
<tagada></tagada> ou <tagada/>
```

D'un certain point de vue, on peut considérer que XML est une description d'arbre. Le code suivant

```
<coordonnees>  
  <adresse>  
    <rue>Jules Verne</rue>  
    <numero>200</numero>  
    <code_postal>80000</code_postal>  
    <ville>Amiens</ville>  
  </adresse>  
  <telephone>03 22 80 81 39</telephone>  
  <email>dep@u-picardie.fr</email>  
</coordonnees>
```

correspond à l'arbre suivant :



Les **balises** qui délimitent les éléments obéissent à des règles d'écriture :

- les balises ne doivent contenir que des chiffres, des lettres, des signes particuliers (il faut éviter les signes ":", ">", "<", "''", "''", "-")
- les balises ne doivent pas commencer par un chiffre, ni un signe particulier
- les balises ne doivent pas comporter d'espace
- les balises ne peuvent commencer par les lettres xml

Quelles sont, ci-dessous, les balises incorrectes et pourquoi ?

```
<nabuchodonosor >  
<joseph dupont>  
<1erprix>  
<joseph_dupont>
```

Comme en HTML, il est possible d'ajouter des **commentaires** dans un code XML (la syntaxe est la même que pour HTML) :

```
<!--ceci est un commentaire-->
```

Les **entités** sont des variables qui représentent des chaînes de caractères (ce qu'on trouve généralement entre une paire de balises). Elles se représentent par un nom précédé du signe "&". Par exemple &rage est une entité qui référence la chaîne de caractères : "O rage, O désespoir, O vieillesse ennemie, N'ai-je donc tant vécu que pour cette infamie et ne suis-je blanchi dans les travaux guerriers que pour voir en un jour flétrir tant de lauriers"

Ainsi

```
<don_diegue>&rage</don_diegue>
```

remplace avantageusement

```
<don_diegue>O rage, O désespoir, O vieillesse ennemie, N'ai-je donc tant vécu que pour cette infamie et ne suis-je blanchi dans les travaux guerriers que pour voir en un jour flétrir tant de lauriers</don_diegue>
```

Nous verrons que les entités se définissent dans les DTD (Document Type Definition).

il peut arriver que l'on veuille insérer dans du code XML des signes "interdits". On peut alors signaler à l'analyseur XML d'ignorer certains passages du code avec une **section CDATA** :

```
<comparaison><![CDATA[huit > trois]]></comparaison
```

permet d'afficher "huit > trois". Sans section CDATA, le signe ">" aurait provoqué une erreur car aurait été analysé comme une fermeture de balise.

Les **attributs** complètent les balises en apportant des informations complémentaires ou indispensables. Dans HTML, les attributs étaient déjà connus ; exemples : src, href,...

```
  
<a href="http://www.ici.fr">
```



## Définition de Type de Document (DTD)

### Éléments

Dans une DTD, on définit principalement les éléments intervenant dans un fichier XML. La définition d'un élément possède la syntaxe suivante :

```
<!ELEMENT nom_élément (modèle_contenu)>
```

nom\_élément est le nom de la balise qui sera utilisée ; modèle\_contenu renseigne sur ce qui est compris entre <nom\_élément> et </nom\_élément>.

Ainsi

```
<!ELEMENT date (jour, mois, année)>
<!ELEMENT jour (#PCDATA)>
<!ELEMENT mois (#PCDATA)>
<!ELEMENT année (#PCDATA)>
```

définit un élément composé, date, contenant trois éléments simples, jour, mois, année. Ces trois éléments ont un contenu de caractères (PCDATA signifie Parsed Character Data ou chaîne de caractères analysée).

un élément vide, comme la balise <hr/> par exemple se définira par emploi du mot réservé EMPTY :

```
<!ELEMENT hr EMPTY>
```

Prenons maintenant un exemple plus complet. Imaginons une bibliothèque composée d'ouvrages. Chaque ouvrage est décrit par un numéro d'inventaire, un titre, un ou plusieurs auteurs, une maison d'édition, un prix d'achat.

Nous avons à construire le fichier XML et la DTD qui définira les balises employées..

Le document XML (enregistré dans le fichier livres.xml) est :

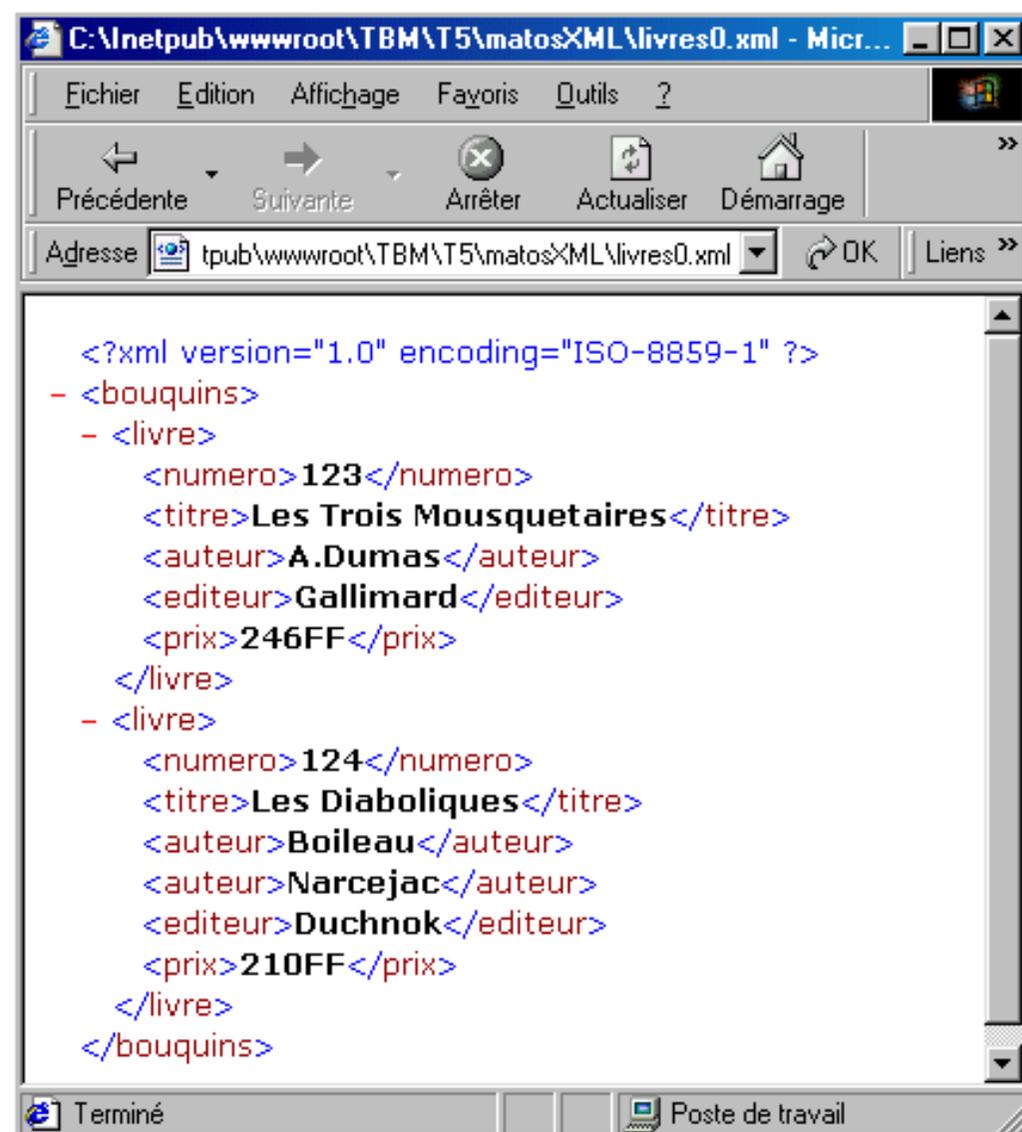
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<bouquins>
  <livre>
    <numero>123</numero>
    <titre>Les Trois Mousquetaires</titre>
    <auteur>A.Dumas</auteur>
    <editeur>Gallimard</editeur>
    <prix>246FF</prix>
  </livre>
  <livre>
    <numero>124</numero>
    <titre>Les Diaboliques</titre>
    <auteur>Boileau</auteur>
```

```

<auteur>Narcejac</auteur>
<editeur>Duchnok</editeur>
<prix>210FF</prix>
</livre>
</bouquins>

```

Les figures suivantes donnent la visualisation du fichier livres.xml dans les navigateurs Netscape6 et Internet Explorer 5 :



Il nous faut maintenant établir la DTD qui définit les balises utilisées. Elle correspondra au fichier suivant enregistré sous le nom de livres.dtd :

```
<!ELEMENT bouquins (livre+)>
<!ELEMENT livre (numero, titre, auteur+, editeur, prix?)>
<!ELEMENT numero (#PCDATA)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT auteur (#PCDATA)>
<!ELEMENT editeur (#PCDATA)>
<!ELEMENT prix (#PCDATA)>
```

Dans une DTD chaque définition de balise est introduite avec la balise <!ELEMENT>. Quelques explications sont nécessaires à la compréhension de ce qui précède :

"livre+" indique que plusieurs occurrences de "livre" vont intervenir. Autrement dit "bouquins" possède plusieurs "livres".

"prix?" indique que le champ "prix" est facultatif : il peut être absent.

Les signes "+", "?" indiquent des contraintes d'occurrence. Il y en a en fait trois :

contrainte	signification
?	0 ou 1 (optionnel)
*	0 ou plus
+	au moins 1

Il faut évidemment indiquer dans le fichier livres.xml quelle DTD il faut utiliser. Pour cela, on rajoute une ligne (en rouge) :

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE bouquins SYSTEM "livres.dtd">
<bouquins>
  <livre>
    <numero>123</numero>
    <titre>Les Trois Mousquetaires</titre>
    <auteur>A.Dumas</auteur>
    <editeur>Gallimard</editeur>
    <prix>246FF</prix>
  </livre>
  <livre>
    <numero>124</numero>
    <titre>Les Diaboliques</titre>
    <auteur>Boileau</auteur>
    <auteur>Narcejac</auteur>
    <editeur>Duchnok</editeur>
    <prix>210FF</prix>
  </livre>
</bouquins>

```

## Attributs

Les composants d'un fichier XML ne comportent pas que des balises ; celles-ci peuvent contenir des attributs. Il faut donc que la DTD puisse expliciter les attributs de balises. Ceci est effectué avec la syntaxe suivante :

```
<!ATTLIST nom_élément nom_attribut type_attribut valeur_par_défaut>
```

nom\_élément est le nom de l'élément auquel est attaché l'attribut ; nom\_attribut est le nom de l'attribut ; type\_attribut est le type de l'attribut ; les différents types sont donnés dans le tableau ci-dessous :

type	signification
CDATA	la valeur est une chaîne de caractères

(val1 val2 .....)	la valeur est l'une des possibilités de l'énumération
ID	la valeur est un identifiant unique
IDREF	la valeur est l'identifiant d'un autre élément
IDREFS	la valeur est une liste d'identifiants
NMTOKEN	la valeur est un nom valide XML
NMTOKENS	la valeur est une liste de noms valides XML
ENTITY	la valeur est une entité
ENTITIES	la valeur est une liste d'entités
NOTATION	la valeur est un nom de notation
xml:	la valeur est une valeur prédéfinie XML

valeur\_par\_défaut peut prendre les valeurs suivantes :

valeur	signification
valeur	valeur par défaut de l'attribut
#REQUIRED	saisie obligatoire
#IMPLIED	saisie facultative
#FIXED	valeur constante prédéfinie

exemples :

dans la DTD	dans le fichier XML
<pre>&lt;!ELEMENT message (#PCDATA)&gt; &lt;!ATTLIST message longueur CDATA "0"&gt;</pre>	<pre>&lt;message longueur="20"&gt;.....&lt;/message&gt;</pre>

<pre>&lt;!ELEMENT paiement (#PCDATA)&gt; &lt;!ATTLIST paiement mode (chèque espèces carte) "espèces"&gt;</pre>	<pre>&lt;paiement mode="carte"&gt;.....&lt;/paiement&gt;</pre>
<pre>&lt;!ELEMENT utilisateur (#PCDATA)&gt; &lt;!ATTLIST utilisateur username CDATA #REQUIRED&gt; &lt;!ATTLIST utilisateur password CDATA #REQUIRED&gt;</pre>	<pre>&lt;utilisateur username="dupont" password="azerty567"&gt;.....&lt;/utilisateur&gt;</pre>

## Entités

Une entité est une variable remplaçant du texte. Elle est définie dans une DTD par la syntaxe

```
<!ENTITY nom_entité valeur_entité>
```

exemples :

```
<!ENTITY souris "mickey mouse">
<!ENTITY texte "il était une fois une marchande de foie qui vendait du foie dans la ville de Foix. Elle me dit,
'ma foi, c'est la première fois que je vends du foie dans la ville de Foix.'">
```

Les entités s'utiliseront dans le document XML sous la forme &souris, &texte.

Les entités précédentes sont des entités internes. On peut aussi définir des entités externes. Leur définition se situe dans un autre document. La syntaxe est alors :

```
<!ENTITY nom_entité SYSTEM "URI">
```

exemple :

```
<!ENTITY Noe SYSTEM "http://www.bible.fr/arche.dtd">
```

On peut aussi définir des entités paramètres. Leur syntaxe est

```
<!ENTITY %nom_entité "type_entité">
```

## exemple

```
<!ENTITY %car "(#PCDATA)">
<!ELEMENT titre %car>
<!ELEMENT sous_titre %car>
```

%car remplace, dans l'exemple précédent (#PCDATA) dans toute la DTD.

Nous avons vu qu'Internet Explorer 5 pouvait agir comme afficheur de structure, mais il ne tient pas compte du contenu de la DTD. Internet Explorer 5 est un "analyseur non validant". Existe-t-il des analyseurs "validants", c'est à dire faisant le lien entre le fichier XML et la DTD ? La réponse est oui. Il en existe un, gratuit, offert par Microsoft (Si ! Ca existe !), appelé msxml.

Pour activer cet analyseur validant, il faut ajouter le code Javascript suivant (que l'on peut placer dans un document HTML) :

```
xmlDoc= new ActiveXObject("msxml2.DOMDocument");
xmlDoc.validateOnParse = true;
xmlDoc.load(url);
```

Information : <http://www.xmlwriter.net/msxml.shtml#UseMSXML>



## Espaces de noms, HTML dans XML, XML dans HTML

### Espaces de noms

Etant donné que les balises sont d'une utilisation libre en XML, il doit fatalement arriver qu'un même nom de balise peut correspondre à des significations différentes. Par exemple la balise <table>, bien connue en HTML, et qui sert à désigner un tableau, pourrait être utilisée en XML, par exemple :

```

<table>
  <tr>
    <td>Athos</td>
    <td>Porthos</td>
    <td>Aramis</td>
  </tr>
</table>

```

Par ailleurs, on pourrait utiliser une balise <table> pour désigner.....une table tout simplement :

```

<table>
  <style>Louis XV</style>
  <forme>rectangulaire</forme>
  <hauteur>1 m</hauteur>
  <largeur>1,50 m</largeur>
  <longueur>2,50 m</longueur>
</table>

```

Imaginons que dans un même document XML, on utilise les deux types de balise <table>. Une confusion est inévitable. Pour l'éviter, on peut utiliser un préfixe qui évite cette confusion :

```

<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Athos</h:td>
    <h:td>Porthos</h:td>
    <h:td>Aramis</h:td>
  </h:tr>
</h:table>
<m:table xmlns:m="http://www.u-picardie.fr">
  <m:style>Louis XV</m:style>
  <m:forme>rectangulaire</m:forme>
  <m:hauteur>1 m</m:hauteur>
  <m:largeur>1,50 m</m:largeur>
  <m:longueur>2,50 m</m:longueur>
</m:table>

```

Dans l'exemple ci-dessous, un attribut xmlns (xml name space) a été ajouté aux balises <h:table> et <m:table> ; cet attribut définit un **espace de noms** (name space) unique par l'intermédiaire d'une URI (Uniform Resource Identifier) qui identifie une ressource Internet de manière unique. Dans la pratique, on peut utiliser comme URI, une URL (Uniform Resource Locator) désignant un domaine Internet ; c'est ce qui a été utilisé dans l'exemple précédent. Peu importe sur quoi pointe l'URL (elle n'est pas censée pointer sur quelque chose de particulier), l'intérêt est de définir de manière unique un espace de noms (évidemment l'URL doit correspondre à un domaine existant).

Il est aussi possible de définir un espace de noms par défaut ; dans ce cas le préfixe est absent :

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Athos</td>
    <td>Porthos</td>
    <td>Aramis</td>
  </tr>
</table>
<m:table xmlns:m="http://www.u-picardie.fr">
  <m:style>Louis XV</m:style>
  <m:forme>rectangulaire</m:forme>
  <m:hauteur>1 m</m:hauteur>
  <m:largeur>1,50 m</m:largeur>
  <m:longueur>2,50 m</m:longueur>
</m:table>
```

Les URI peuvent être déclarées de manière absolue ou relative :

déclaration absolue : <http://www.u-picardie.fr/~cochard/sim> (adresse complète)

déclaration relative : ~cochard/sim

Pour interpréter les URI relatives, on utilise **XMLBase** :

```
<tagada xml:base="http://www.u-picardie.fr/">
  <.....href="~cochard/sim"...../>
</tagada>
```

Dans l'exemple ci-dessus, la valeur de l'attribut href est <http://www.u-picardie.fr/~cochard/sim>

Il est possible d'intégrer du code HTML dans un document XML et réciproquement.

## Intégration de HTML dans XML

Supposons que l'on souhaite intégrer dans un fichier XML le code HTML suivant :

```
<hr/>
<h1>Attention</h1>
<p>Ce texte est issu d'un fichier en code HTML</p>
<hr/>
```

Dans le document XML, par exemple celui du début de ce chapitre, on définit un espace de noms :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<document xmlns:html="http://www.w3.org/TR/REC-html40">
  <question>Bonjour</question>
  <html:hr/>
  <html:h1>Attention</html:h1>
  <html:p>Ce texte est issu d'un fichier en code HTML</html:p>
  <html:hr/>
  <reponse>Salut</reponse>
</document>
```



Cette méthode est très générale et permet d'intégrer au sein de XML, des codes relevant de divers langages à balises.

### Intégration de XML dans HTML

La procédure inverse existe aussi. Il faut créer dans le document HTML un **ilot de données** XML compris entre les balises `<xml id=nom_de_l'_ilot>` et `</xml>`. Par exemple, soit le fichier HTML suivant :

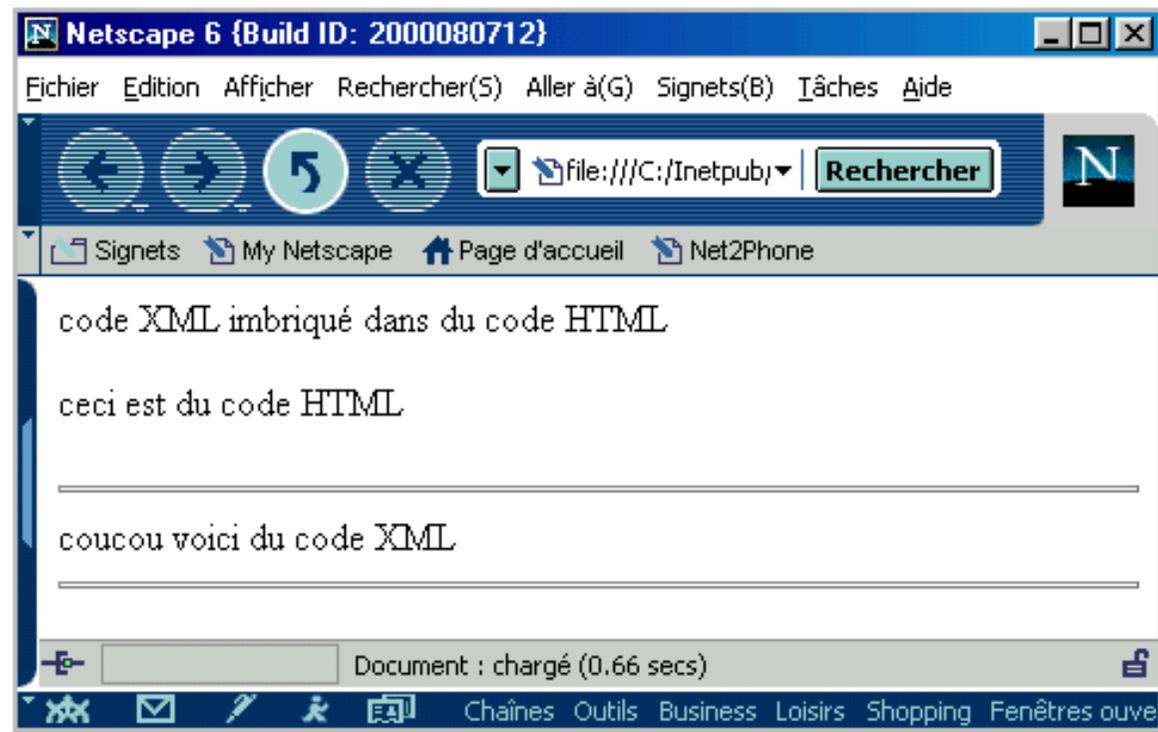
```
<html>
  <head>
    <titre>code XML imbriqué dans du code HTML</titre>
  </head>
  <body>
    <p>ceci est du code HTML</p>
    <hr/>
    <hr/>
  </body>
</html>
```

Intercalons du code XML :

```
<tagada>
  <tsoin>coucou</tsoin>
  <tralala>voici du code XML</tralala>
</tagada>
```

ce qui donne :

```
<html>
  <head>
    <titre>code XML imbriqué dans du code HTML</titre>
  </head>
  <body>
    <p>ceci est du code HTML</p>
    <hr/>
    <xml id="ilotXml">
      <tagada>
        <tsoin>coucou</tsoin>
        <tralala>voici du code XML</tralala>
      </tagada>
    </xml>
    <hr/>
  </body>
</html>
```



## Mise en page : CSS

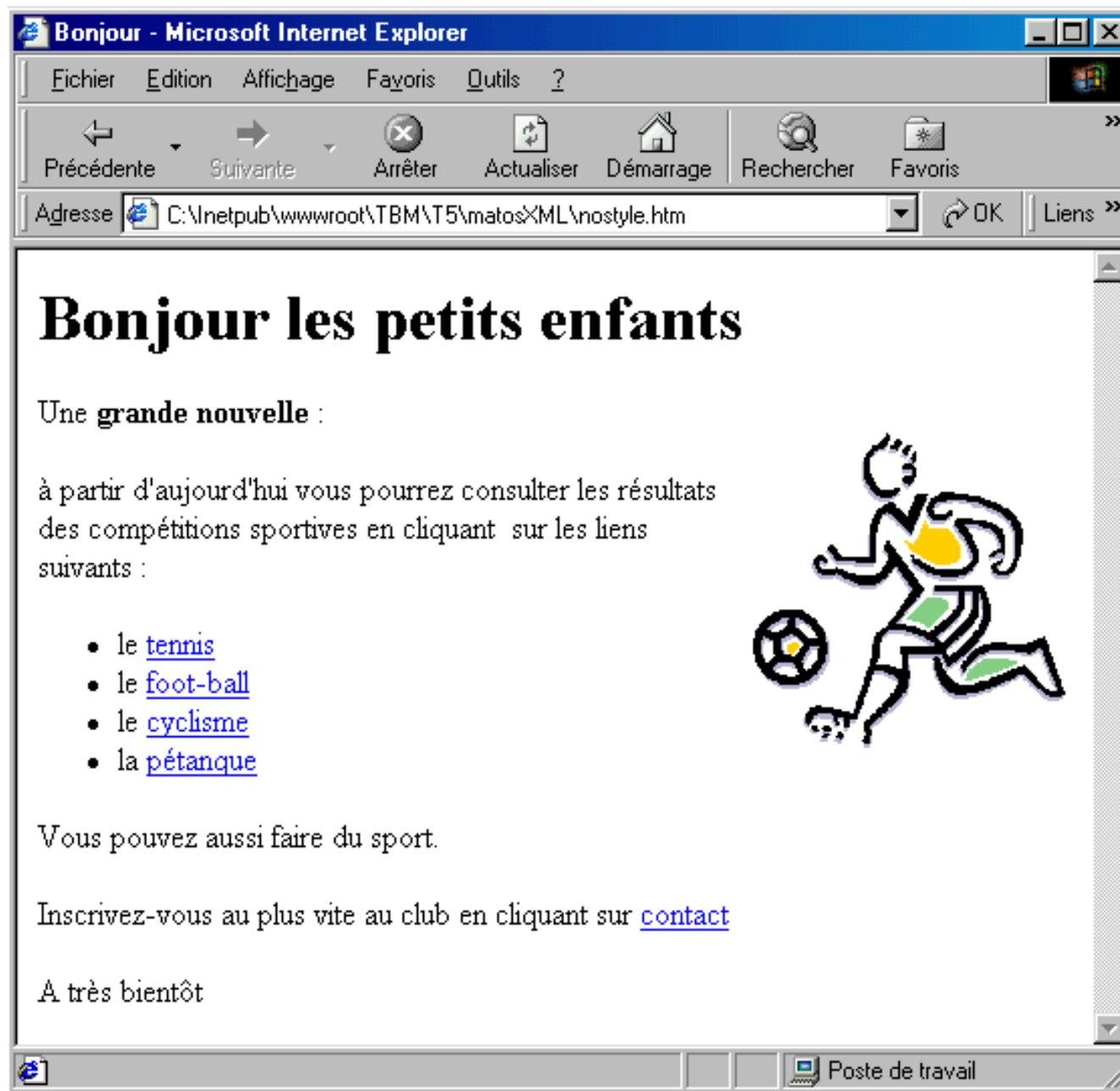
Comme nous l'avons vu, XML ne se préoccupe que de la structure des documents, pas de la mise en page. Pour mettre en page un document XML, deux possibilités sont offertes : CSS et XSL. Nous étudions ici la première possibilité.

Cascading Style Sheets (CSS) permet de définir des feuilles de style qui accompagnent un document (HTML ou XML). Une feuille de style, comme indique son nom, définit les styles qui sont utilisés dans un document. Elle correspond elle-même à un document reconnaissable à son suffixe .css.

Pour faire comprendre leur utilisation, nous utiliserons un exemple en HTML . Soit le document HTML suivant :

```
<html>
  <head><title>Bonjour</title></head>
  <body>
    <h1>Bonjour les petits enfants</h1>
    <p>Une <b>grande nouvelle</b> :</p>
    
    <p>à partir d'aujourd'hui vous pourrez consulter les résultats des compétitions sportives
      en cliquant sur les liens suivants :</p>
    <ul>
      <li>le <a href="tennis.htm"> tennis</a></li>
      <li>le <a href="foot.htm"> foot-ball</a></li>
      <li>le <a href="velo.htm"> cyclisme</a></li>
      <li>la <a href="cochonnet.htm"> pétanque</a></li>
    </ul>
    <p>Vous pouvez aussi faire du sport.</p>
    <p>Inscrivez-vous au plus vite au club en cliquant sur
    <a href="mailto:zidane@u-picardie.fr">contact</a></p>
    <p>A très bientôt</p>
  </body>
</html>
```

Le résultat dans Internet Explorer est donné ci-dessous :



Nous allons joindre une feuille de style que nous appellerons style1.css à cette page. Tout d'abord, il faut indiquer dans le document HTML où se trouve la feuille de style. Cela se fait simplement en ajoutant la ligne

```
<link rel="stylesheet" type="text/css" href="style1.css"/>
```

dans la section <head></head>

Ensuite il faut rédiger la feuille de style sous forme d'un document texte (style1.css) :

```
body {margin: 1.0cm;}
h1 {font-family: Comic Sans MS, Helvetica, sans-serif; font-size: 30pt;}
p {font-family: Arial, Helvetica, sans-serif; font-size: 15pt;}
b {color: red;}
ul {font-family: Times New Roman, Helvetica, sans-serif; font-size : 18pt;}
a:link {color: blue;}
a:hover {color: red;}
a:active {color: green;}
```

où l'on peut constater que, pour chaque balise, le style est défini. En particulier, on remarquera que pour les polices de caractères, il est utile d'en définir plusieurs car l'utilisateur ne possède pas forcément sur sa machine les polices que vous utilisez.

Avec l'ajout de la feuille de style, le résultat est donné ci-dessous :



à partir d'aujourd'hui vous pourrez consulter les résultats des compétitions sportives en cliquant sur les liens suivants :

- le [tennis](#)
- le [foot-ball](#)
- le [cyclisme](#)
- la [pétanque](#)



Vous pouvez aussi faire du sport.

Inscrivez-vous au plus vite au club en cliquant sur [contact](#)

A très bientôt

Terminé

Poste de travail

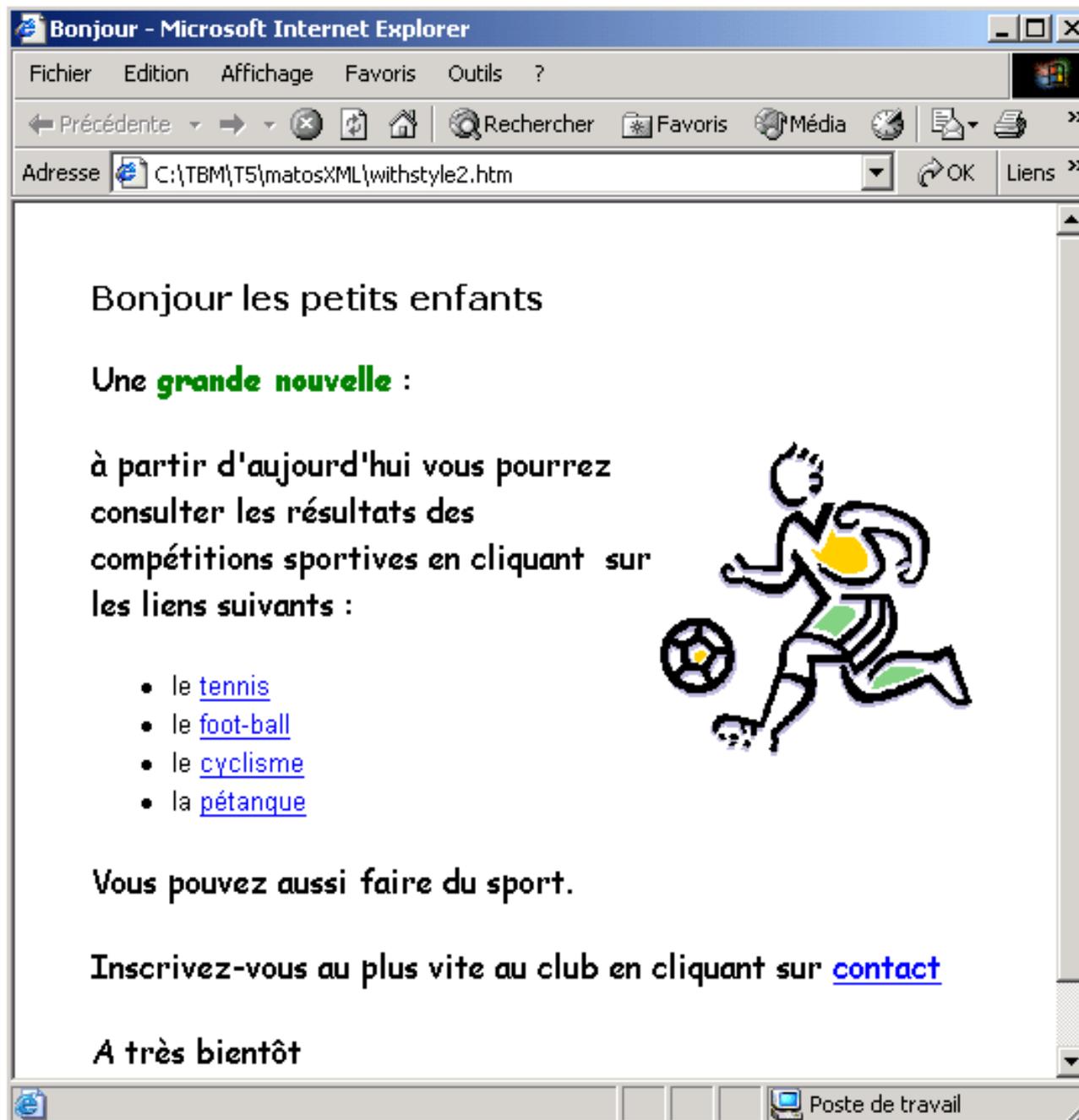
Imaginons maintenant une autre feuille de style, style2.css

```
body {margin: 1.0cm;}
h1 {font-family: Verdana, Helvetica, sans-serif; font-size: 20pt;}
p {font-family: Comic Sans MS, Helvetica, sans-serif; font-size: 20pt;}
b {color: green;}
ul {font-family: Arial, Helvetica, sans-serif; font-size : 18pt;}
a:link {color: blue;}
a:hover {color: red;}
a:active {color: green;}
```

Cette feuille peut être ajoutée en déclaration dans le fichier HTML, en plus de la feuille de style, style1.css.

```
<html>
  <head><title>Bonjour</title>
    <link rel="stylesheet" type="text/css" href="style1.css"/>
    <link rel="stylesheet" type="text/css" href="style2.css"/>
  </head>
  <body>
    <h1>Bonjour les petits enfants</h1>
    <p>Une <b>grande nouvelle</b> :</p>
    
    <p>à partir d'aujourd'hui vous pourrez consulter les résultats des compétitions sportives
    en cliquant sur les liens suivants :</p>
    <ul>
      <li>le <a href="tennis.htm"> tennis</a></li>
      <li>le <a href="foot.htm"> foot-ball</a></li>
      <li>le <a href="velo.htm"> cyclisme</a></li>
      <li>la <a href="cochonnet.htm"> pétanque</a></li>
    </ul>
    <p>Vous pouvez aussi faire du sport.</p>
    <p>Inscrivez-vous au plus vite au club en cliquant sur
    <a href="mailto:zidane@u-picardie.fr">contact</a></p>
    <p>A très bientôt</p>
  </body>
</html>
```

On obtient le résultat suivant :



Cet exemple nous apprend que l'on peut superposer les feuilles de style css, d'où le nom "cascading", les dernières directives d'affichage prenant le pas sur les directives plus anciennes.

Qu'en est-il pour un document XML. C'est à peu près la même chose. Reprenons l'exemple des livres (livres.xml) donné plus haut. Intercalons le lien avec la feuille de style appelée livres.css (en rouge en deuxième ligne) ; soit livres2.xml, le fichier correspondant :

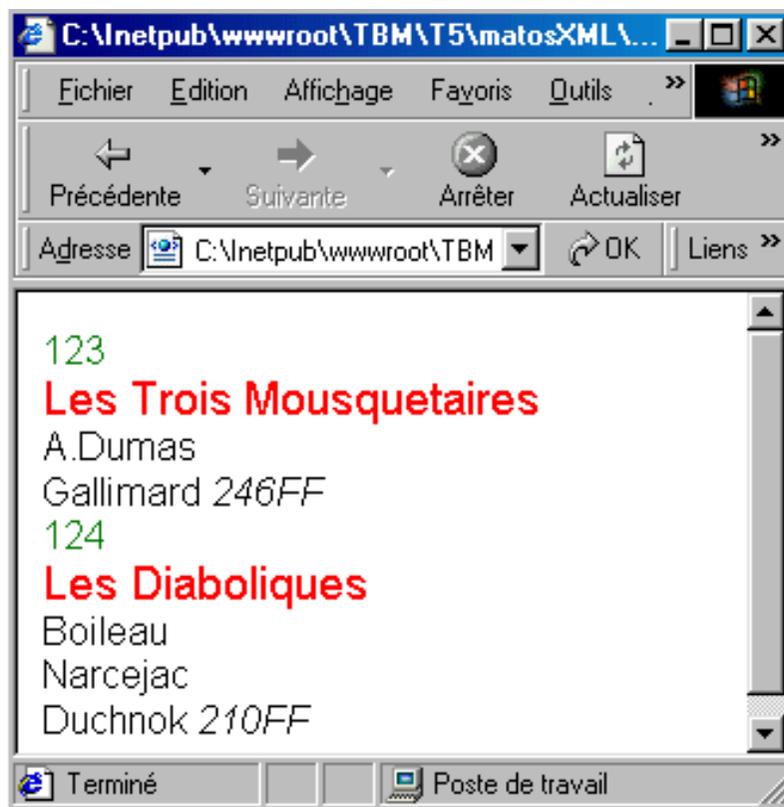
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet href="livres.css" type="text/css"?>
<bouquins>
  <livre>
    <numero>123</numero>
    <titre>Les Trois Mousquetaires</titre>
    <auteur>A.Dumas</auteur>
    <editeur>Gallimard</editeur>
    <prix>246FF</prix>
  </livre>
  <livre>
    <numero>124</numero>
    <titre>Les Diaboliques</titre>
    <auteur>Boileau</auteur>
    <auteur>Narcejac</auteur>
    <editeur>Duchnok</editeur>
    <prix>210FF</prix>
  </livre>
</bouquins>
```

La feuille de style doit maintenant être établie suivant un mode opératoire assez similaire à celui utilisé plus haut : pour chaque balise on définit le style :

```
bouquins {font-family : Arial, Helvetica, sans-serif;}
numero {display: block; font-size : 12pt; color=green;}
titre {display: block; font-size: 14pt; color: red; font-style: bold }
auteur {display: block; font-size: 12pt}
editeur { display: inline; font-size : 12pt;}
prix {display: inline; font-size :12pt; font-style: italic;}
```

L'attribut display est particulièrement utile. Il permet d'afficher quelque chose, puis, soit de passer à la ligne suivante [\(voir la page suivante\)](#)

l'affichage correspond à un bloc) , soit de continuer sur la même ligne ("inline"). On notera aussi que la police de caractère étant définie dans "bouquins" est commune aux éléments "intérieurs". Le résultat est donné ci-dessous dans Internet Explorer :



Un aspect intéressant de CSS concerne les images d'arrière plan. Imaginons que l'on souhaite ajouter une image d'arrière-plan à l'affichage précédent. Voici quelques possibilités

<p>l'image est répétée régulièrement mais seulement sur l'élément bouquins.</p>	<pre> bouquins {font-family : Arial, Helvetica, sans-serif; background-image: url(photo.gif)} numero {display: block; font-size : 12pt; color=green;} titre {display: block; font-size: 14pt; color: red; font- style: bold } auteur {display: block; font-size: 12pt} editeur { display: inline; font-size : 12pt;} prix {display: inline; font-size :12pt; font-style: italic;} </pre>	
<p>l'image est répétée régulièrement mais seulement sur l'élément éditeur</p>	<pre> bouquins {font-family : Arial, Helvetica, sans-serif; } numero {display: block; font-size : 12pt; color=green;} titre {display: block; font-size: 14pt; color: red; font- style: bold } auteur {display: block; font-size: 12pt} editeur { display: inline; font-size : 12pt; background- image: url(photo.gif)} prix {display: inline; font-size :12pt; font-style: italic;} </pre>	

<p>L'image n'est pas répétée et est positionnée en haut à gauche sur l'élément bouquins</p>	<pre>bouquins {font-family : Arial, Helvetica, sans-serif; background-image: url(photo.gif); background-repeat: no-repeat} numero {display: block; font-size : 12pt; color=green;} titre {display: block; font-size: 14pt; color: red; font- style: bold } auteur {display: block; font-size: 12pt} editeur { display: inline; font-size : 12pt;} prix {display: inline; font-size :12pt; font-style: italic;}</pre>	
<p>L'image est centrée horizontalement et verticalement sur l'élément bouquins</p>	<pre>bouquins {font-family : Arial, Helvetica, sans-serif; background-image: url(photo.gif); background-repeat: no-repeat ; background-position: center center} numero {display: block; font-size : 12pt; color=green;} titre {display: block; font-size: 14pt; color: red; font- style: bold } auteur {display: block; font-size: 12pt} editeur { display: inline; font-size : 12pt;} prix {display: inline; font-size :12pt; font-style: italic;}</pre>	

CSS a fait l'objet d'une standardisation en 1996 sous le nom CSS1. Depuis une nouvelle version CSS2 a vu le jour en 1998 et encore une autre, CSS3, est en préparation. Mais les navigateurs usuels ne reconnaissent encore que CSS1.



## Mise en page : XSL

XSL signifie Extensible Style Sheet Language. Dans sa version de base, il exploite les balises de HTML (qui ne sont pas si mauvaises que cela après tout !) mais au bénéfice de XML. Il peut aussi utiliser une mise en forme spécifique basée sur les FO (Formatting Objects ou Objets de formatage). En fait XSL est bien plus qu'un langage de feuille de style car il peut

- transformer les documents XML (le plus souvent en XHTML) : cette partie s'appelle XSLT
- filtrer et trier les données XML : cette partie s'appelle XPath
- formater les données XML pour une restitution à l'écran, sur papier,... : cette partie est basée sur les FO

XSL est un standard défini et recommandé par le W3 Consortium. Pour voir les résultats de XSL, il faut employer un navigateur compatible XSL. Internet Explorer 5 n'est pas totalement compatible avec XSL ; par contre Internet Explorer 6 l'est ; on supposera donc dans la suite que l'on utilise Internet Explorer 6.

Contrairement à CSS, XSL correspond à un fichier qui doit être "inséré" dans un document XML ; c'est pourquoi, on peut considérer le contenu d'un fichier XSL comme un espace de noms. Pour expliquer l'utilisation de XSL, reprenons l'exemple du début du chapitre :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<document>
  <question>Bonjour</question>
  <reponse>Salut</reponse>
</document>
```

et ajoutons en deuxième ligne :

```
<?xml-stylesheet href="bonjour.xsl" type="text/xsl"?>
```

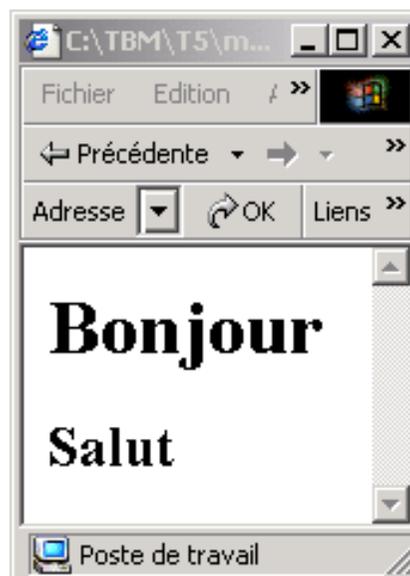
Le fichier bonjour.xsl aura la physionomie suivante :

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <h1>
      <xsl:value-of select="document/question"/>
    </h1>
    <h2>
      <xsl:value-of select="document/reponse"/>
    </h2>
  </xsl:template>
</xsl:stylesheet>
```

La première ligne correspond à la définition correcte (au sens de la conformité avec la recommandation du W3C) d'une feuille de style XML, donc utilisable avec Internet Explorer 6. Si on utilise Internet Explorer 5, il faut mettre la ligne suivante :

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

Dans l'exemple ci-dessus, nous voyons que les balises de HTML sont ici réutilisées pour des portions de document. Le résultat sera le suivant :



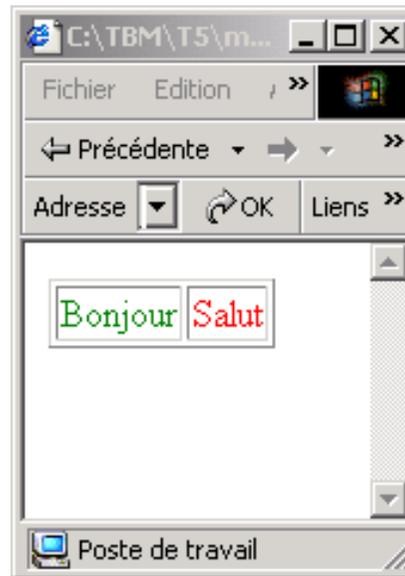
On peut changer la présentation en utilisant le même fichier xml, mais un fichier xsl différents, par exemple bonjour2.xml :

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <table border="1">
      <tr>
        <td><font color="green">
          <xsl:value-of select="document/question"/>
        </font>
        </td>
        <td><font color="red">
          <xsl:value-of select="document/reponse"/>
        </font>
        </td>
      </tr>
    </table>
  </xsl:template>
</xsl:stylesheet>
```

Bien entendu, on placera dans le fichier `bonjour.xml` la ligne suivante :

```
<?xml-stylesheet href="bonjour2.xsl" type="text/xsl"?>
```

Le résultat sera alors :



Contrairement à CSS, XSL utilise des modèles (templates) associés à des éléments d'un fichier XML. L'association d'un modèle à un élément s'opère avec la balise `<xsl:template>` et avec l'attribut "match". Dans l'exemple ci-dessus, la ligne

```
<xsl:template match="/">
```

indique que le modèle défini entre les balises "template" s'applique à l'élément racine "/". Les éléments à présenter sont définis à l'aide de la balise `<xsl:value-of>`

```
<xsl:value-of select="document/question"/>
```

L'attribut "select" indique la position de l'élément dans la structure.

L'utilisation de XSL permet de faire un véritable traitement du texte contenu dans un fichier XML. Pour illustrer ces possibilités, reprenons l'exemple des livres vu au chapitre sur CSS et appelons biblio.htm le fichier suivant qui comprend la description de trois ouvrages :

```
<?xml version="1.0"?>
<?xml-stylesheet href="biblio.xsl" type="text/xsl"?>
<bouquins>
  <livre>
    <numero>123</numero>
    <titre>Les Trois Mousquetaires</titre>
    <auteur>A.Dumas</auteur>
    <editeur>Gallimard</editeur>
    <prix>60 Euros</prix>
  </livre>
  <livre>
    <numero>124</numero>
    <titre>Les Diaboliques</titre>
    <auteur>Boileau</auteur>
    <auteur>Narcejac</auteur>
    <editeur>Duchnok</editeur>
    <prix>78 Euros</prix>
  </livre>
  <livre>
    <numero>125</numero>
    <titre>Cinq semaines en ballon</titre>
    <auteur>Jules Verne</auteur>
    <editeur>Hetzel</editeur>
    <prix>inestimable</prix>
  </livre>
</bouquins>
```

Imaginons que nous souhaitions faire la liste des livres avec mention du numéro, des auteurs, du titre, de l'éditeur.

Le fichier XSL, biblio.xsl pourra s'écrire ainis :

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<table border="1">
  <xsl:for-each select="bouquins/livre">
    <tr>
      <td><xsl:value-of select="numero"/></td>
      <td>
        <xsl:for-each select="auteur">
          <table border="0">
            <tr>
              <td><xsl:value-of select="."/></td>
            </tr>
          </table>
        </xsl:for-each>
      </td>
      <td><xsl:value-of select="titre"/></td>
      <td><xsl:value-of select="editeur"/></td>
    </tr>
  </xsl:for-each>
</table>
</xsl:template>
</xsl:stylesheet>
```

Le résultat, affiché avec Internet Explorer 6 sera :

123	A.Dumas	Les Trois Mousquetaires	Gallimard
124	Boileau Narcejac	Les Diaboliques	Duchnok
125	Jules Verne	Cinq semaines en ballon	Hetzel

On utilise dans l'exemple ci-dessus des boucles, définies par la balise "for-each" :

```
<xsl:for-each select="nom_de_noeud">
```

et on notera en particulier l'imbrication des boucles de ce type (qui permet de faire afficher plusieurs auteurs pour le même livre).

# Gestion de l'information multimédia

## Le Multimédia et sa spécificité

---

### Objectifs

1. Spécificité des données multimédias
2. Spécificité des supports
3. Spécificité des outils et des méthodes de développement
4. Spécificité des outils de traitement
5. Spécificité des outils de diffusion
6. Bibliographie

 Téléchargement du chapitre (chap3.pdf - ... Ko)

---

Concepteurs : 

Gérard-Michel Cochard

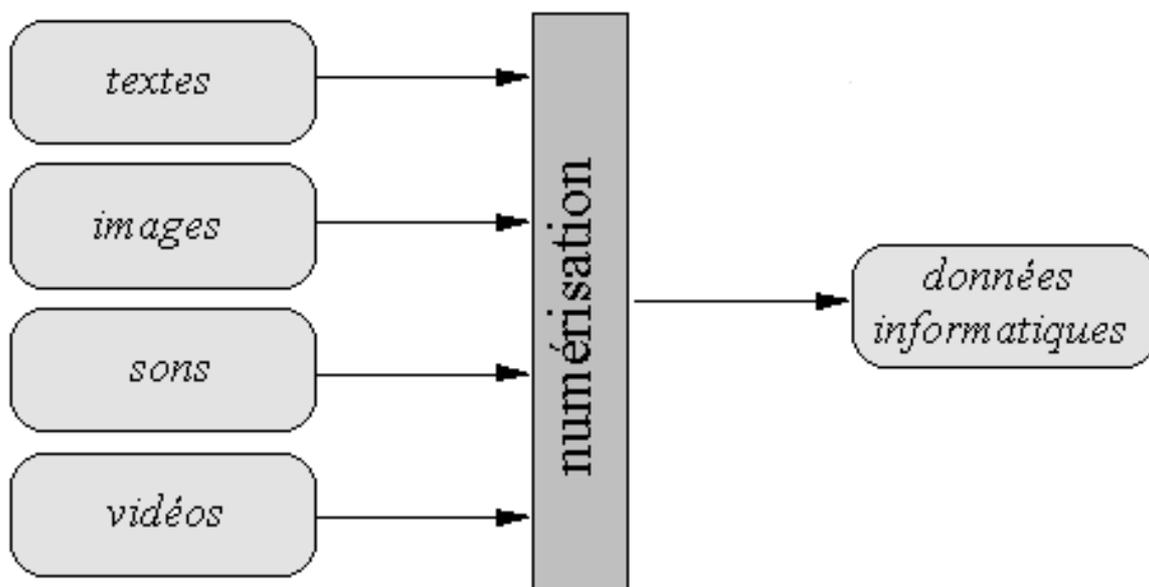
# 1 - Spécificité des données multimédias



Le multimédia est l'intégration au sein d'un même système d'information de données textuelles, d'images fixes ou animées, de sons, de vidéo. Son histoire est très récente, depuis 1990 environ, et va de pair avec la tendance prononcée à la numérisation de l'information, la production de petits systèmes informatiques à possibilités très performantes et au développement de l'interconnexion de réseaux, notamment dans ce dernier cas, Internet et le World Wide Web.

Selon les points de vue, le multimédia recouvre un marché en pleine expansion, un environnement de travail et de loisir, des applications professionnelles ou grand public, une industrie portant sur la production de matériels et logiciels. Une définition possible est celle-ci (selon Terrasson) : le multimédia est l'exploitation simultanée de données sonores, visuelles, informatiques, l'ensemble des techniques de création, de stockage, de transmission, de restitution, et aussi l'ensemble des données permettant cette exploitation simultanée.

Un premier intérêt du multimédia est l'intégration généralisée des données et des outils de manipulation de ces données. Avant le multimédia, des appareillages divers étaient nécessaires : un ordinateur pour les données informatiques, un magnétophone pour les cassettes de sons, un magnétoscope pour les vidéos, un téléphone pour le transport de la voix, du papier pour les images, ...L'utilisation simultanée de données de natures différentes nécessitait donc une juxtaposition inconfortable de ces outils. L'utilisation de l'ordinateur comme outil unique est un progrès sensible. Par ailleurs, les données multimédias sont manipulées comme des données informatiques puisque codées sous forme de 0 et de 1 et peuvent donc, en particulier, être stockées de manière unique et transmises de manière unique ce qui constitue un second intérêt



Cette intégration se fait cependant avec quelques problèmes :

- le volume de données numérisées est généralement important, ce qui nécessite des techniques de compression/décompression.
- Les traitements sont plus complexes et nécessitent des matériels et logiciels de plus en plus performants
- la transmission de données volumineuses nécessitent des débits binaires sur les réseaux qui commencent seulement à apparaître.

Néanmoins cette tendance semble irréversible et entraîne une reconfiguration du paysage autrefois audio-visuel en paysage totalement numérique.

La plupart des informations sont distribuées sous forme de textes, d'images fixes, de sons, de vidéo. Actuellement, il est possible de numériser ces médias (il n'est cependant pas encore possible de numériser les odeurs, les goûts et les sensations de toucher).

## Les textes

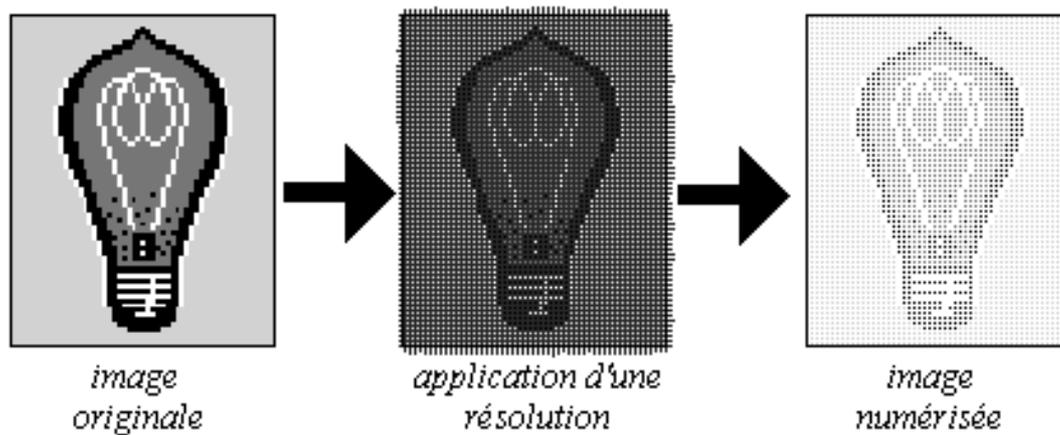
Composés de caractères dits d'imprimerie, la numérisation s'opère simplement par codage de chaque caractère en une suite de 0 et de 1. Le code ASCII (American Standard Code for Information Interchange) sur 7 bits permet de coder 128 caractères usuels. Comme les ordinateurs travaillent usuellement sur des mots qui sont des multiples de mots de 8 bits (octets), on peut rajouter un 0 devant le code ASCII, ce qui correspond au code normalisé. Ainsi le caractère "A" est codé 01000010. On peut aussi rajouter un 1 et définir un second jeu de 128 caractères (code ASCII étendu) ; malheureusement ce second jeu n'est pas normalisé et varie d'une plate forme informatique à l'autre.

Il est également possible de coder la mise en forme du texte : gras, italique, souligné, la taille des caractères, l'alignement, etc. Là encore, il n'y a pas de normalisation et les logiciels manipulant du texte possèdent encore leur propre système.

Pour un texte comportant 2000 caractères (espaces compris), sans style, la taille du fichier numérisé sera donc de 2000 octets ce qui correspond à un volume faible.

## Les images fixes

Pour les applications multimédias, les images sont converties en matrices de points. Chaque point est codé suivant sa "couleur" et les codes résultants sont placés séquentiellement dans un fichier, ligne par ligne, colonne par colonne.



Pour une image binaire, c'est à dire "bicolore", noir et blanc, chaque point est codé sur 2 bits (0 pour noir, 1 pour blanc. Pour une image avec 256 nuances de gris allant du noir total au blanc total, chaque point sera codé sur 8 bits (de 00000000 pour noir à 11111111 pour blanc). Pour une image couleur, on exprime cette couleur comme une superposition d'une dose de Rouge, d'une dose de Vert et d'une dose de Bleu (système RVB) ; si on permet une variation de 0 à 255 pour une dose, une couleur quelconque sera codée comme la juxtaposition de trois octets, soit 24 bits (et donc 16 777 216 couleurs possibles).

Considérons une image pour laquelle la grille de codage se compose de 640\*480 points. Si l'image est binaire, sa taille est de 38 400 octets, si l'image est à 256 nuances de gris, sa taille est de 307 200 octets ; si l'image possède des couleurs codées sur 24 bits, sa taille est de 921 600 octets. Si l'on considère qu'une disquette contient usuellement 1 400 000 octets, on comprend que ce support est quelque peu limité pour le stockage de nombreuses images. Par ailleurs, des techniques de compression permettent de gagner de la place mais au prix d'une décompression au moment de l'affichage des images ; ce que l'on gagne en volume est perdu en vitesse d'affichage.

La taille du fichier obtenu après numérisation et avant compression est fortement influencée par le pas de la grille de codage qui est appelé résolution et exprimé usuellement en points par pouce (dpi : dots per inch). Pour une même image une résolution de 600 dpi donnera un fichier image 36 fois plus volumineux que le fichier image résultant d'une numérisation à 100 dpi.

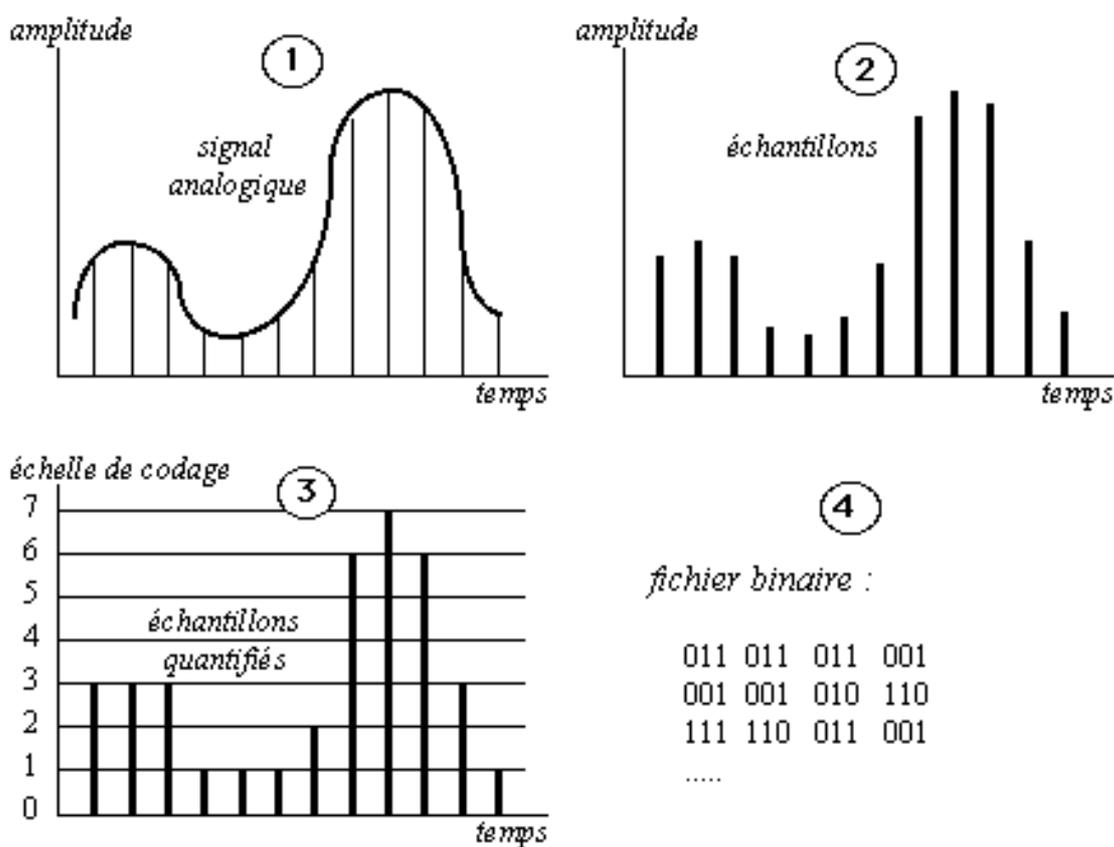
Les techniques de compression conduisent à des formats d'images, soit normalisés, soit imposés par les industriels. Deux formats d'image sont couramment employés pour les applications multimédias en ligne : le format GIF qui correspond à des images compressées sans perte d'information avec 256 nuances de couleurs et le format JPEG qui correspond à des images compressées avec perte d'information avec 16 millions de couleurs. Ces deux formats ont l'immense avantage d'être interprétés par la plupart des navigateurs du Web.

## Les sons

Les technologies d'acquisition du son ont été longtemps analogiques : le son était représenté par les variations d'une grandeur physique, une tension électrique par exemple. Les techniques actuelles

permettent d'obtenir directement un son numérisé : c'est notamment le cas des magnétophones produisant un enregistrement sur cassettes D.A.T. (Digital Audio Tapes, digitalisation du son à une fréquence de 48 KHz) .

Pour numériser un son enregistré de manière analogique, on procède en trois étapes :



1. Échantillonnage : l'amplitude du signal analogique est mesurée à une fréquence d'échantillonnage  $f$ . On obtient ainsi une collection de mesures.
2. Quantification : une échelle arbitraire allant de 0 à  $2^n - 1$  est employée pour convertir les mesures précédentes. Une approximation est faite de manière à ce que chaque mesure coïncide avec une graduation de l'échelle (cette approximation, qui modifie légèrement le signal, est appelée bruit de quantification).
3. Codage : suivant sa grandeur dans cette nouvelle échelle, chaque mesure est codée sur  $n$  bits et placée séquentiellement dans un fichier binaire.

On comprendra que les valeurs de  $f$  et de  $n$  sont critiques pour la taille du fichier résultant.

Usuellement 3 qualités de numérisation sont employées :

- la qualité Hifi ou CD audio :  $f=44$  KHz,  $n=16$  bits, stéréo (2 signaux sonores)
- la qualité "radio" :  $f= 22$  KHz,  $n=8$  bits, mono ou stéréo
- la qualité "téléphonique" :  $f=11$  KHz,  $n=8$  bits, mono

Ainsi, 1 heure de son correspondra à des fichiers de 630 Mo, 158 Mo et 40 Mo pour les qualités CD, radio et téléphonique respectivement. Nous verrons plus loin que les standards de CD ROM

correspondent à une capacité d'environ 640 Mo. Les normes de codage numérique du son portent sur l'enregistrement non compressé des mesures quantifiées (PCM : Pulse Code Modulation), sur l'enregistrement des différences entre deux mesures successives, ce qui permet une réduction de volume de données (DPCM : Delta PCM), ou encore sur l'utilisation de techniques de prédiction des mesures (ADPCM : Adaptative Differential PCM).

## La vidéo

Comme pour le son, la vidéo s'exprime par des signaux de nature analogique qu'il faut numériser pour une utilisation multimédia. Il existe ici quatre signaux à numériser : trois pour l'image et un pour le son. Les signaux d'image se répartissent en signal de luminance Y et deux signaux de chrominance Db et Dr, reliés aux composantes R,V,B (Rouge, Vert, Bleu) par les relations linéaires

$$Y = 0.30R + 0.59V + 0.11B$$

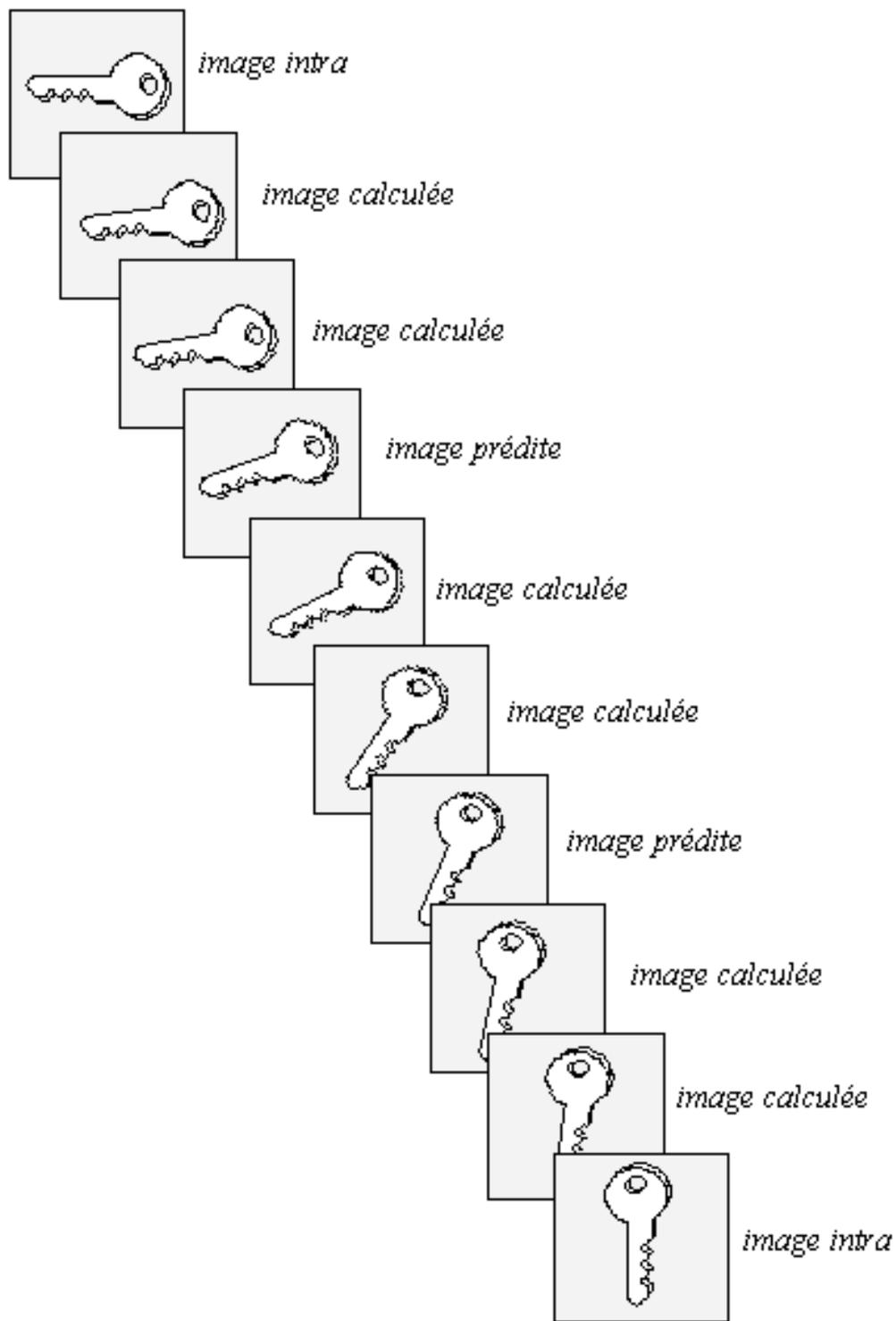
$$Db = B - Y$$

$$Dr = R - Y.$$

La composante Y est échantillonnée à une fréquence de 13,5 MHz, tandis que les composantes de chrominance sont échantillonnées à une fréquence de 6,75 MHz. La quantification pour les trois signaux d'image s'opère sur 8 bits.

Les données peuvent être simultanément ou ultérieurement compressées. Plusieurs systèmes sont actuellement utilisés. La compression Vidéo d'Intel est utilisée dans le logiciel Vidéo For Windows (le résultat de la numérisation et de la compression conduit à des fichiers vidéo AVI). Pour des plates-formes Macintosh, mais aussi Windows, le logiciel Quick Time d'Apple joue un rôle analogue à partir de fichiers vidéo MOV ; un grand intérêt de Quick Time est sa possibilité d'intégration de données initiales diverses (images, sons, textes) et de proposer plusieurs types de compression. Les logiciels Vidéo For Windows et Quick Time réalisent également la synchronisation du son avec les images.

La norme MPEG, spécialement élaborée pour la compression de vidéo, tend actuellement à s'imposer. Le principe de compression s'appuie sur trois types d'images : les images "intra" sont des images peu compressées qui servent de repère (une image intra pour 10 images successives) ; les images "prédites" sont des images obtenues par codage et compression des différences avec les images intra ou prédites précédentes (une image prédite toutes les trois images) ; les images "interpolées" sont calculées comme images intermédiaires entre les précédentes. L'utilisation de vidéos numériques MPEG nécessite la présence d'une carte de décompression dans le micro-ordinateur d'exploitation.

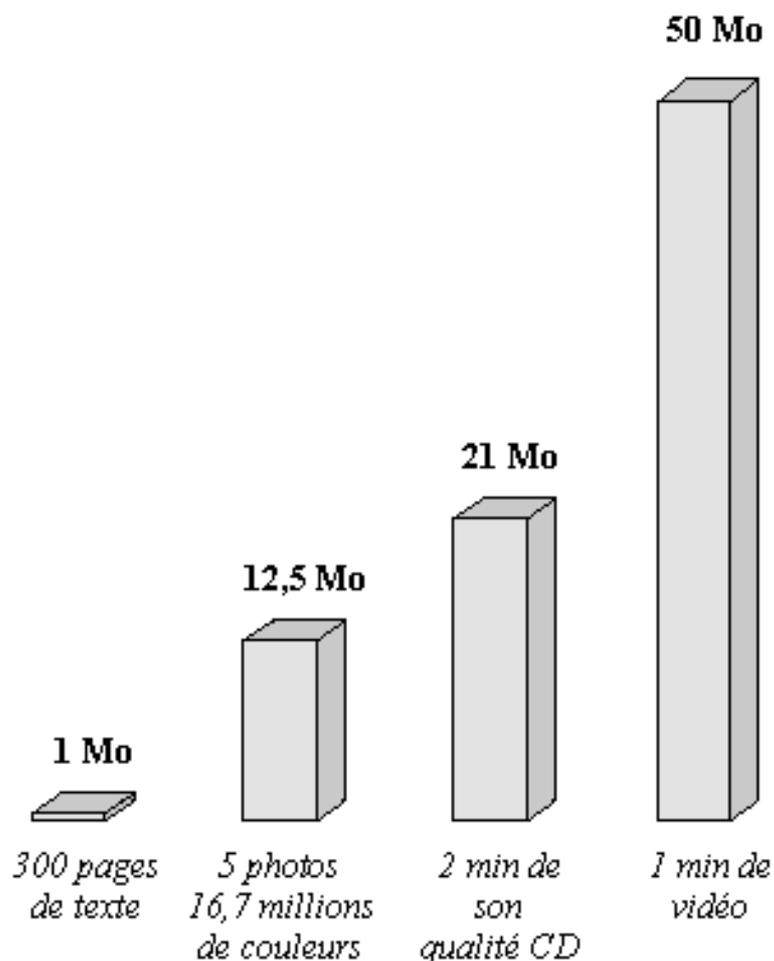


## 2 - Spécificité des supports



Les supports des données ou des applications multimédia doivent satisfaire deux contraintes essentielles :

- posséder des capacités importantes de stockage
- posséder des taux d'entrée/sortie suffisants pour l'exploitation en temps réel.



Les supports classiques que sont les disquettes 1,4 Mo sont évidemment à éliminer du fait de leur faible capacité. Les disques durs magnétiques pourraient être de bons candidats pour l'hébergement d'une application multimédia de taille raisonnable ; en particulier, leur débit d'entrée/sortie est satisfaisant ; mais leurs coûts et leurs technologies qui les rendent indissociables de l'ordinateur (peu adapté au stockage) les éliminent, quant il s'agit de supports de diffusion commerciale, au profit de la famille des CD-ROM qui, née avec le multimédia, est actuellement le support quasi-universel des applications et des données multimédias distribués off line.

Par contre, pour des applications mettant en jeu des bases de données, accessibles en local ou à distance, l'utilisation des disques durs est encore la meilleure solution, sauf, peut-être, pour des données dont les modifications sont rares ; dans ce dernier cas, l'utilisation de dispositifs de type juke-box de CD-ROM peut être une solution.



## 3 - Spécificité des outils et méthodes de développement



Le développement d'une application multimédia implique des outils spécifiques adaptés au type des données à traiter ou à leur intégration.

Les outils qui sont liés au type de donnée sont

- pour l'image : outils d'acquisition, de dessin, d'édition et de retouche.
- pour le son : outils de numérisation, d'édition et de retouche
- pour la vidéo : outils de numérisation, d'édition et de retouche, d'animation
- pour le texte : outils d'édition, de reconnaissance optique de caractères.

Suivant les cas, ces outils sont des composants matériels et/ou logiciels.

Les outils d'intégration sont logiciels et correspondent soit à la structuration des données, soit à leur manipulation et en particulier à la réalisation de l'interface utilisateur. Les outils de structuration des données sont principalement des systèmes de gestion de bases de données dotés d'extensions permettant la gestion de données multimédias. Les outils de manipulation peuvent être des langages auteurs orientés multimédias (ils réalisent alors directement l'intégration de données multimédias de types différents) ou des langages de programmation usuels (c'est au programmeur de réaliser tous les traitements).



## 4 - Spécificité des outils de traitement



Le multimédia nécessite, pour son exploitation, des ressources supplémentaires par rapport à celles des ordinateurs classiques. La station multimédia est un ordinateur, le plus souvent micro-ordinateur, doté

- de composants matériels performants: processeurs rapides (200 MHz), mémoire vive de grande capacité (32 Mo), écran 640x480 au moins ;
- d'extensions spécifiques : carte son, baffles ou casque, microphone, carte vidéo, carte réseau, modem, lecteur de CD-ROM,... ;
- de modules logiciels particuliers pour l'exploitation du son, de la vidéo, de l'image, de la communication (notamment navigateur www).

Heureusement ces ressources se sont généralisées et les micro-ordinateurs du marché sont de plus en plus multimédias dans leur configuration de base et les prix sont au-dessous de 20 000 F pour la quasi-totalité des constructeurs.

Toutefois ces considérations ne s'appliquent qu'aux "postes" d'exploitation ; les dispositifs de stockage et de diffusion que constituent les serveurs peuvent être, au contraire, d'une très grande complexité technique et d'un coût important.



## 5 - Spécificité des outils de diffusion



Dans le domaine du multimédia, on parle souvent d'applications "on line" et d'application "off line" pour indiquer que l'exploitation des données multimédias s'effectue en connexion avec un réseau informatique ou en local sans réseau. Plus précisément, ces deux modes d'exploitation correspondent aux deux modes de diffusion actuels que sont la diffusion par CD-ROM et la diffusion par le World Wide Web.

Le CD-ROM est le moyen de stockage le plus pratique pour le multimédia et son utilisation première a été pour l'enregistrement du son sur le CD Audio en remplacement du disque vinyle. D'une capacité globale de 640 Mo, le CD-ROM permet l'enregistrement de 200 000 pages de texte brut, ou de 2000 images en 256 couleurs, ou de 250 images en 16 millions de couleurs, ou de 12 minutes de vidéo, ou de 60 minutes de son, ou, bien sûr, un mixage de différents médias. Ces chiffres montrent à la fois les avantages et les inconvénients du CD-ROM : sa capacité est grande mais ses limites sont vite atteintes. L'apparition récente du DVD (Digital Versatile Disk) permet de multiplier par 10 les capacités de stockage ; malheureusement le parc existant n'est pas équipé du lecteur spécifique de DVD et l'impact de ce nouveau support de diffusion n'est pas encore mesurable.

L'autre mode de diffusion consiste en le composant Internet appelé World Wide Web (WWW ou Web). Basé sur une architecture de type client-serveur ce service assure la délivrance par un serveur sur un poste client de documents multimédias encodés par un langage de description appelé HTML (HyperText Mark up Language). Ces documents HTML sont interprétés par un logiciel spécifique, le navigateur, installé sur le poste client. Le navigateur (browser, en anglais) permet la communication avec le serveur et l'affichage du document. Le protocole HTTP (HyperText Transfer Protocol) qui régit les communications entre le serveur et le client est l'outil de base pour la diffusion sur réseau de documents multimédias. On connaît le succès de cette technologie qui correspondait en 1996 à 25% du trafic sur réseau en France.



## 6 - Bibliographie



J. TERRASSON	Les outils du multimédia	éditions Armand Colin, 1992
J-M. HERELLI ER	Le Multimédia	Sybex
T. VAUGHAN	Multimedia, making it work	Mc Graw Hill
E. HOLSI NGER	Le Multimédia...Comment ça marche ?	Dunod



# Gestion de l'information multimédia

## Bases de Données

---

### Objectifs

#### N.B

1. Organisation et modélisation
2. Modèle relationnel
3. Un peu de méthodologie
4. Modèle orienté objet
5. Produits à tendance multimédia
6. Bibliographie

 Téléchargement du chapitre (chap4.pdf - ... Ko)

---

Concepteurs : 

Gérard-Michel Cochard

# NB :

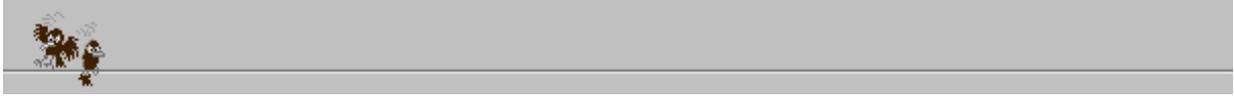


Ce chapitre s'adresse aux non spécialistes des bases de données. Il a un but d'initiation et de sensibilisation (en anglais "primer"). Il ne saurait donc constituer une référence exhaustive sur le sujet et il n'en a pas l'ambition.

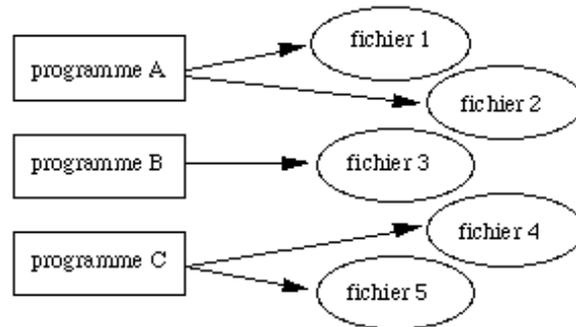
---

⇒ Sections du chapitre : [1](#) | [2](#) | [3](#) | [4](#) | [5](#) | [6](#)

# 1 - Organisation et modélisation

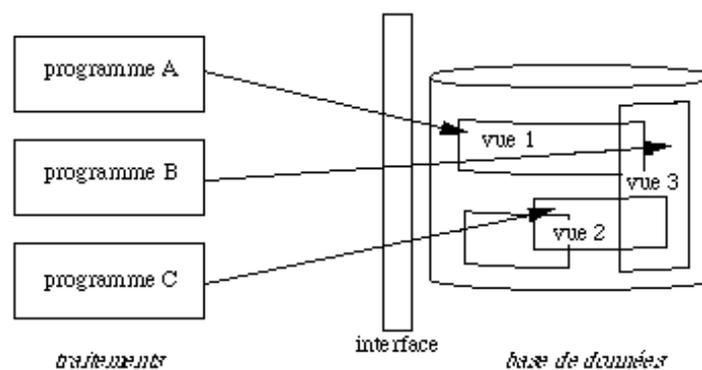


Les premières applications informatiques, dans le domaine de la gestion, consistaient en tandems programmes-fichiers. Les fichiers contenaient les données nécessaires aux traitements opérés par les programmes. Le tandem programmes-fichiers était indissociable car les fichiers ne pouvaient être manipulés que par les programmes et les programmes ne pouvaient s'exécuter sans leurs fichiers. Comme les applications étaient développées successivement avec des matériels différents et des langages de programmation divers, toutes ces applications n'étaient pas communicantes et étaient juxtaposées. Une mise à jour des données devenait alors très complexe.



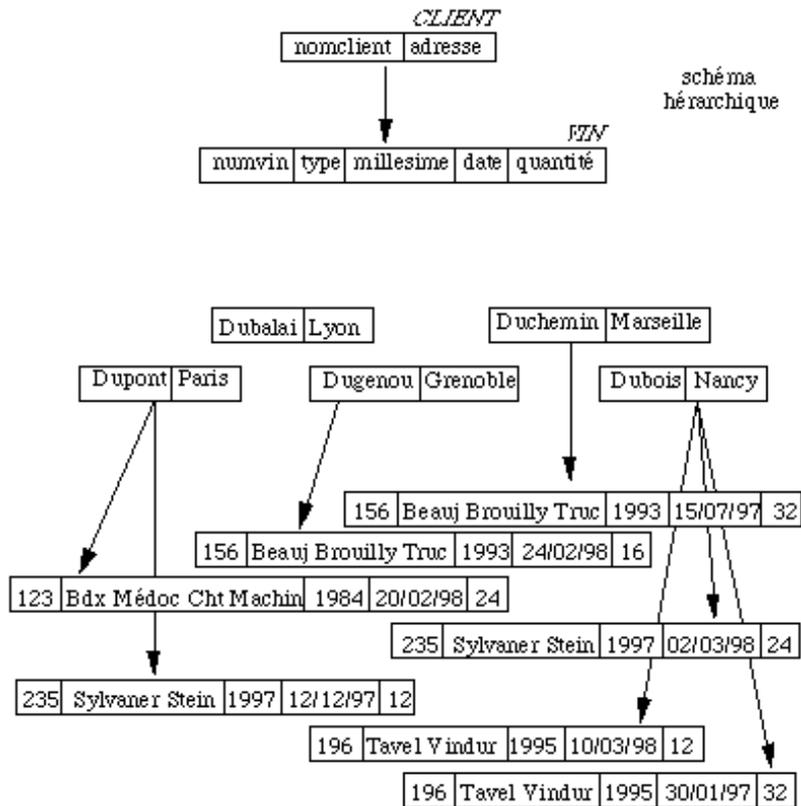
C'est alors que le concept de séparation des traitements et des données donna naissance aux bases de données dont une définition peut être celle d'une collection structurée de données enregistrées sur des supports accessibles par l'ordinateur et partagées par plusieurs utilisateurs .

De fait, dans une base de données, ces dernières forment un tout gérable indépendamment des traitements par un système de gestion de base de données qui permet de structurer, d'administrer et de sécuriser la base. Les traitements peuvent être créés indépendamment des données et, par le biais d'une interface appropriée, accéder à une partie (vue) de la base de données. L'essor des bases de données est dû également à la technologie des disques magnétiques de grande capacité à accès direct.

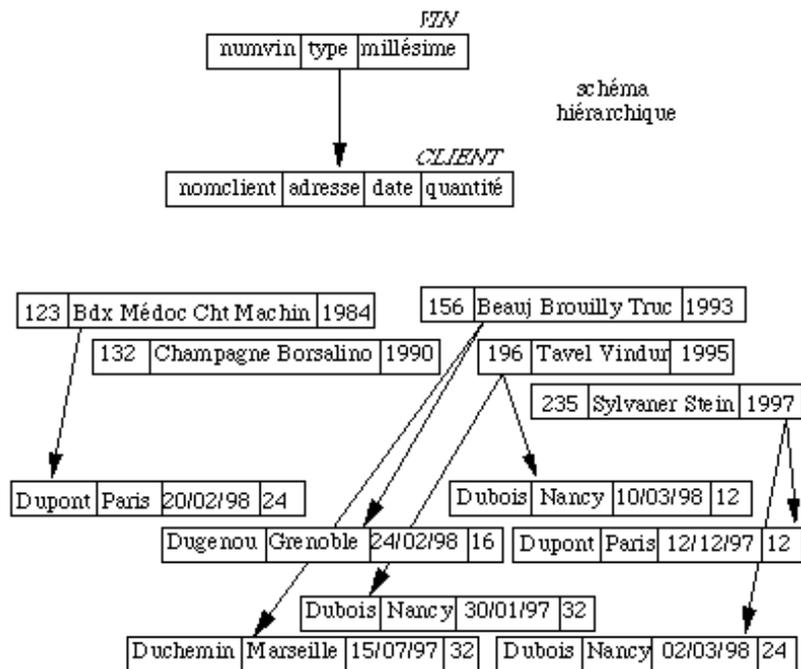


Les premières bases de données (1965-1970) se sont appuyées sur des modèles appelés "navigationnels" étant donné le mode d'accès aux données. Le modèle hiérarchique, premier à être utilisé, notamment avec le produit IMS (Information Management System), structurait les données sous forme d'arbres, l'entrée dans la base se faisant par la racine ; le modèle qui a suivi fut le modèle "réseau" ou CODASYL qui, avec la notion de connecteur, facilita les exécutions de requêtes mais au prix d'une gestion par pointeurs assez lourde.

Exemple : Supposons qu'une maison de négoce en vins répertorie ses clients dans une base de données faisant apparaître les produits commercialisés (numéro de vin, type, millésime, ...), et les clients (nom, prénom,) ainsi que les achats de ces clients : numéro de vin acheté, date d'achat, nombre de bouteilles. Le modèle hiérarchique peut correspondre au schéma client à vin (modèle H1) dont quelques occurrences sont données ci-dessous.



On pourrait d'ailleurs imaginer aussi un modèle hiérarchique vin à client (modèle H2, figure ci-dessous).



Quel que soit le modèle hiérarchique adopté, les requêtes peuvent être quelquefois complexes dans leurs réalisations. Ainsi, si dans le modèle H1, on recherche les vins achetés par le client "Dubois", la requête s'exprimera simplement :

```

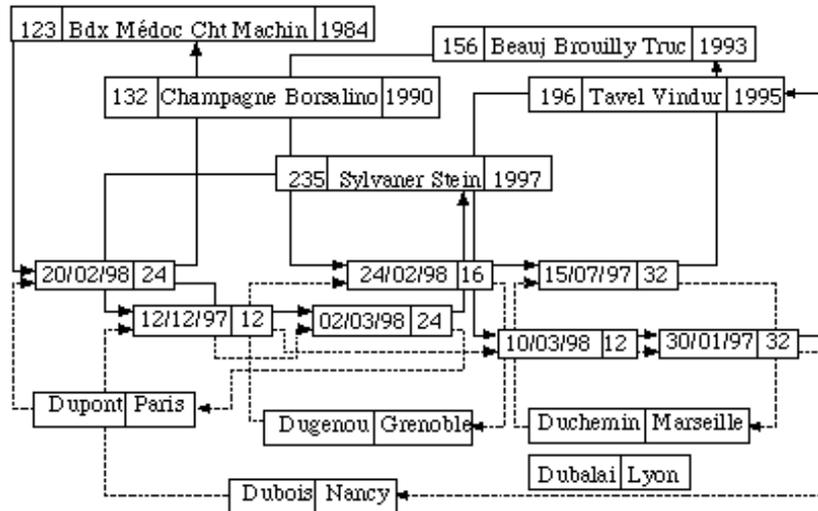
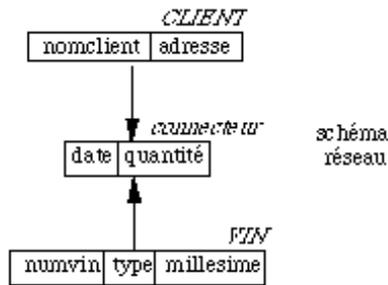
Rq1 :   Trouver client "Dubois"
        Tant qu'il y a des vins sous "Dubois"
            Trouver n°vin
            Imprimer n°vin
        Fin Tant Que
    
```

Si, par contre, on recherche les clients qui ont acheté le vin numéro "154", la requête est plus compliquée :

```

Rq2 :   Tant qu'il y a des clients
        Tant qu'il y a des vins sous le client
        Lire numéro de vin sous le client
        Si numéro="154" alors
            imprimer le nom du client
        Fin Si
        Fin Tant Que
    Fin Tant Que
    
```

Le modèle réseau apporte une amélioration en "symétrisant" l'arborescence du modèle hiérarchique par adjonction du connecteur "date achat, nombre de bouteilles". Les associations entre l'entité VIN et le connecteur, d'une part, et entre l'entité CLIENT et le connecteur, d'autre part, correspondent à deux familles de pointeurs (organisés en boucles généralement comme indiqué sur la figure ci-dessous).



Les deux requêtes précédentes s'expriment alors de la manière suivante :

```

Rq1 : Trouver client "Dubois"
        Tant qu'il y a des connecteurs
            Trouver le n°vin associé au connecteur
            Imprimer n°vin
        Fin Tant Que

Rq2 : Trouver le vin numéro "154"
        Tant qu'il y a des connecteurs sous ce vin
            Trouver nom du client associé au connecteur
            Imprimer nom du client
        Fin Tant Que
    
```

On notera la symétrie de ces deux requêtes. On notera aussi les nombreux pointeurs nécessaires à l'implémentation de la base de données réseau ce qui constitue un inconvénient majeur de ce type de modélisation.

## 2 - Modèle relationnel



Le modèle relationnel, défini par Codd au début des années 70, a ses bases dans la théorie mathématique des ensembles et constitue actuellement le modèle de données le plus répandu parmi les produits commercialisés (Oracle, Sybase, Informix, ...). Le concept de base est celui de relation qui n'est autre qu'un ensemble d'éléments appelés "tuples", chacun de ces tuples étant constitué d'attributs prenant leurs valeurs dans un domaine spécifique.

Exemple : En reprenant l'exemple précédent, on peut imaginer la relation suivante

VIN (numvin, type, mil)

où numvin (numéro de vin) prend ses valeurs dans l'ensemble des entiers positifs ; type est un libellé figurant dans une liste définie ; mil (millésime) prend ses valeurs dans l'ensemble des dates (années). Bien entendu, le triplet numvin, type, mil doit avoir un sens. Par exemple le vin (169, Bordeaux Pauillac Château Latour, 2378) n'existe pas encore, il faudra attendre la récolte de l'année 2378.

De même, on peut imaginer la relation

CLIENT(nomclient, prenomclient, adresse, da, nbout)

où da et nbout désignent respectivement la date d'achat et le nombre de bouteilles. Les tables correspondantes aux relations VIN et CLIENT sont données ci-dessous.

## VIN

numvin	type	mil
123	Bdx Médoc Cht Machir	1984
132	Champagne Borsalino	1990
156	Beauj Brouilly Truc	1993
196	Tavel Vindur	1995
235	Sylvaner Stein	1997

## CLIENT

nomclient	prenomclient	adresse	da	nbout
Dupont	Arthur	Paris	20/02/98	24
Dupont	Arthur	Paris	12/12/97	12
Dubois	Gédéon	Nancy	30/01/97	32
Dubois	Gédéon	Nancy	2/03/98	24
Dubois	Gédéon	Nancy	10/03/98	12
Dugenou	Tiburce	Grenoble	24/02/98	16
Duchemin	Léonce	Marseille	15/07/97	32
Dubalai	Stanislas	Lyon		

Les tables ne constituent cependant pas une base de données relationnelle. Pour cela les tables/relations doivent satisfaire certains critères qui sont basés sur le concept de dépendance fonctionnelle.

On dira qu'entre un attribut A et un attribut B il existe une dépendance fonctionnelle A à B si la donnée de A détermine uniquement B.

Exemple : Dans l'exemple précédent, dans la relation VIN, on pourra considérer que le numéro de vin numvin correspond à un vin unique (type) et à un millésime unique (mil). Dans la relation CLIENT, il n'y a pas de dépendance fonctionnelle nomclient à prenomclient car il peut exister deux clients de même nom avec des prénoms différents.

Il n'y a pas de dépendances fonctionnelles évidentes de da vers nbout.

Une dépendance fonctionnelle est **élémentaire** si elle est du type A à B ou si la dépendance (A, B) à C ne peut se réduire à A à C ou B à C.

Exemple : La dépendance fonctionnelle (numvin, type) à mil n'est pas élémentaire car on a la dépendance numvin à mil. Les dépendances fonctionnelles numvin à mil, numvin à type sont élémentaires.

Une dépendance fonctionnelle A à B est **directe** s'il n'existe pas de transitivité du type A à C et C à B. Pour la construction des relations, on ne prend en considération que les dépendances fonctionnelles élémentaires et directes et le signe "à" désignera désormais ce type de dépendance

fonctionnelle.

Si, dans une relation, un attribut A est tel que A à X où X est un attribut quelconque de la relation, alors on définit A comme une clé.

Pour que les relations constituent une base de données relationnelle, elles doivent satisfaire à la condition de normalisation suivante : Les seules dépendances fonctionnelles figurant dans une relation sont celles du type

clé à autres  
attributs.

Exemple : La relation VIN est normalisée puisque numvin est une clé et que numvin à type et numvin à mil :

VIN(numvin, type,  
mil)

La clé est usuellement soulignée soulignée.

La relation CLIENT n'est pas normalisée car il n'existe pas de clé. Pour normaliser cette relation il faut la transformer :

- a. on ajoute un attribut numclient (numéro de client) pour désigner un client de manière unique ; on a alors les dépendances fonctionnelles numclient à nomclient, numclient à prenomclient et numclient à adresse.
- b. il n'y a pas de dépendance fonctionnelle entre numclient et da ou nbout. On les retire de la relation CLIENT qui devient alors CLIENT(numclient, nomclient, prenomclient, adresse).
- c. On est alors conduit à créer une relation supplémentaire ACHAT(numachat, da, nbout, #numvin, #numclient) par adjonction d'un nouvel attribut numachat qui jouera le rôle de la clé de manière à avoir les dépendances fonctionnelles numachat à da, numachat à nbout, numachat à numvin, numachat à numclient. Les signes # devant numvin et numclient indiquent, par convention, que numvin et numclient sont des clés d'autres relations (clés externes).

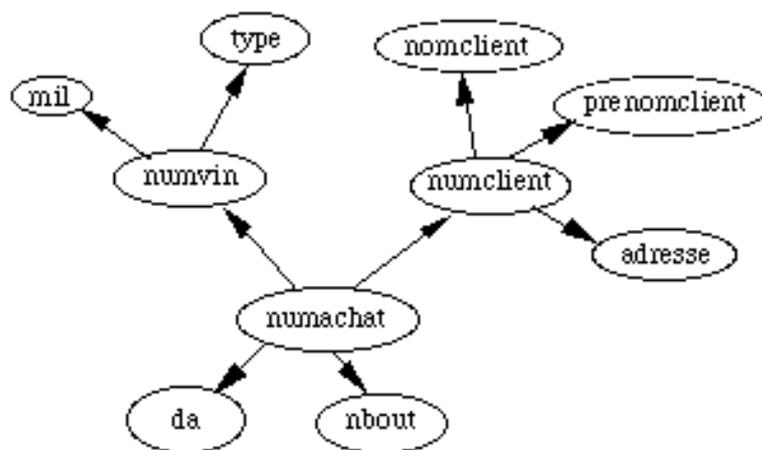
En définitive, on obtient la base de données relationnelle suivante :

VIN(numvin, type, mil)  
CLIENT(numclient, nomclient, prenomclient, adresse)  
ACHAT(numachat, da, nbout, #numvin, #numclient)

VIN			CLIENT			
numvin	type	mil	numclient	nomclient	pre nomclient	adresse
123	Bdx Médoc Cht Machir	1984	1	Dupont	Arthur	Paris
132	Champagne Borsalino	1990	2	Dubois	Gédéon	Nancy
156	Beauj Brouilly Truc	1993	3	Dugenou	Tiburce	Grenoble
196	Tavel Vindur	1995	4	Duchemin	Léonce	Marseille
235	Sylvaner Stein	1997	5	Dubalai	Stanislas	Lyon

ACHAT				
numachat	da	nbout	numvin	numclient
10	30/01/97	32	196	2
11	15/07/97	32	156	4
12	12/12/97	12	235	1
13	20/02/98	24	123	1
14	24/02/98	16	156	3
15	2/03/98	24	235	2
16	10/03/98	12	196	2

Le graphe des dépendances fonctionnelles permet de retrouver les trois relations ci-dessus.. Les attributs sont représentés par des ovales et les flèches indiquent les dépendances fonctionnelles directes et élémentaires.



Un intérêt important des bases de données relationnelles est qu'il existe un langage normalisé pour formuler les requêtes : SQL.

Une requête s'exprime sous la forme d'une phrase type :

```

SELECT nom_attribut1, nom_attribut2, ...
FROM nom_table
WHERE condition ;

```

nom\_attribut1, nom\_attribut2, ...sont les noms des attributs que l'on recherche ; nom\_table est le nom de la table dans laquelle on recherche ; condition définit les critères de recherche.

Exemple : Pour obtenir le nom du client qui possède un numéro de client égal à 1587, la requête SQL s'exprimera comme suit :

```
SELECT nomclient FROM CLIENT WHERE
numclient="1587" ;
```

Il est possible de faire des requêtes sur plusieurs tables et ceci de plusieurs manières comme l'illustre l'exemple suivant.

Exemple : En reprenant les deux requêtes Rq1 et Rq2 de notre exemple introductif, on aura par exemple :

```
Rq1 :  SELECT numvin
        FROM CLIENT, ACHAT
        WHERE (CLIENT.numclient=ACHAT.numclient) AND (CLIENT.nomclient="Dubois") ;
```

```
Rq2 :  SELECT nomclient
        FROM CLIENT, ACHAT
        WHERE (ACHAT.numvin="154") AND (ACHAT.numclient=CLIENT.numclient)
```

On remarquera que lorsqu'un nom d'attribut est le même pour deux tables différentes, on fait précéder le nom de l'attribut par le nom de la table pour éviter toute ambiguïté. On notera également la possibilité d'exprimer des conditions avec des "et logiques" (AND) ou "ou logiques" (OR). On aurait pu aussi obtenir le même résultat en exprimant les requêtes Rq1 et Rq2 sous forme de requêtes en cascade :

```
Rq1 :  SELECT numvin
        FROM ACHAT
        WHERE numclient IN (SELECT numclient
                            FROM CLIENT
                            WHERE nomclient="Dubois") ;
```

```
Rq2 :  SELECT nomclient
        FROM CLIENT
        WHERE numclient IN (SELECT numclient
                            FROM ACHAT
                            WHERE numvin="154") ;
```



## 3 - Un peu de méthodologie



### 3.1 - Modèle entité-association

La conception d'une base de données s'intègre dans la conception plus générale d'un système d'information (voir plus loin) ; celle-ci doit être la plus rigoureuse possible et, à cet effet, s'appuie généralement sur une méthode de conception et d'analyse dont la plus connue, historiquement, en France est la méthode Merise. Cette méthode effectue d'emblée la séparation de l'analyse des traitements de celle des données et procède par niveaux, ce qui, dans la formulation standard de la méthode correspond à 6 modèles (en fait les versions récentes de Merise comportent 8 modèles).

	<b>données</b>	<b>traitements</b>
<b>niveau conceptuel</b>	MCD Modèle Conceptuel des Données	MCT Modèle Conceptuel des Traitements
<b>niveau organisationnel</b>	MLD Modèle Logique des Données	MOT Modèle Organisationnel des Traitements
<b>niveau opérationnel</b>	MPD Modèle Physique des Données	MOpT Modèle Opérationnel des Traitements

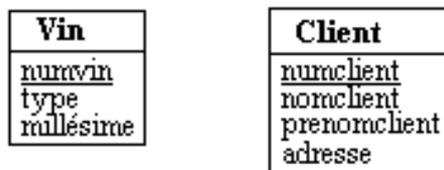
Très grossièrement le niveau conceptuel correspond à la définition de ce que l'on veut obtenir et à la modélisation de la réalité (Quoi ? Que veut-on faire ?), le niveau organisationnel exprime en structures, en processus et en postes de travail, la manière dont le système d'information sera organisé (Qui fait quoi ?, Où ?, Quand ?, Comment ?), le niveau opérationnel est le niveau d'implémentation (Avec quels moyens ?).

Nous n'insisterons pas ici sur les traitements pour ne pas nous éloigner des objectifs de ce cours. Par contre, nous pouvons examiner de plus près le cas des données et en particulier le MCD (Modèle Conceptuel des Données) et le MLD (Modèle Logique des Données).

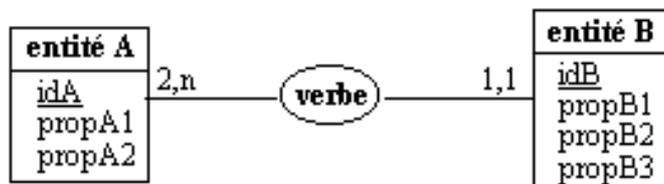
De nos jours, le MCD est exprimé le plus souvent dans le formalisme entité-association basé sur les concepts fondamentaux d'entité et d'association entre entités.

Une entité est supposée modéliser un objet concret ou abstrait du monde réel ; l'entité est repérée sans ambiguïté par un identifiant unique (numéro, nom, matricule, ...). On peut considérer l'entité comme un type générique auquel se rattachent des instanciations ou occurrences. Un certain nombre de propriétés caractérisent l'entité.

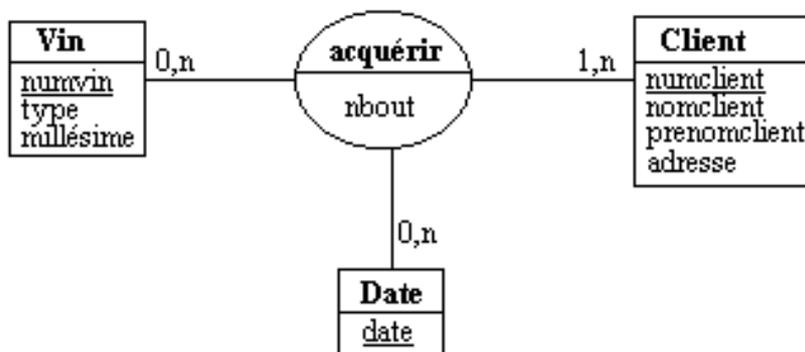
Exemple : Dans l'exemple de la maison de négoce en vin, on peut facilement identifier les entités Vin et Client :



Une association est une liaison entre entités ; elle est usuellement désignée par un verbe, une cardinalité et possède, elle-aussi, éventuellement, des propriétés. Les cardinalités sont pour chaque extrémité de l'association de la forme x, y où x et y désignent respectivement le nombre minimal et le nombre maximal d'occurrences de l'entité "lointaine" qui correspondent à l'occurrence "proche". Dans la figure ci-dessous qui représente une association entre deux entités A et B, la cardinalité 2,n indique qu'à une occurrence de A correspond au minimum 2 occurrences de B et au maximum n (indéfini) occurrences de B ; la cardinalité 1,1 signifie que à 1 occurrence de B correspond une et une seule occurrence de A.



Exemple : Dans le cas de la maison de négoce, on peut définir entre les entités Vin et Client et une nouvelle entité Date, l'association "acquérir" :



Les cardinalités de la figure ci-dessus s'expliquent de la manière suivante : pour un vin donné, il peut y avoir 0 acquisition (ce vin n'a pas de succès) ou plusieurs acquisitions ; un client donné peut acquérir au cours du temps entre 1 vin (sinon il ne serait pas client) et plusieurs vins ; à une date données, plusieurs acquisitions peuvent en général se produire, mais il peut aussi ne pas y en avoir

du tout. L'association précédente est appelée ternaire car elle a trois "pattes" ; en général, on évite, lorsque cela est possible, ce genre d'association ainsi que celles d'ordre supérieur en essayant d'avoir le maximum d'associations binaires.

Lorsque l'on a défini toutes les entités et toutes les associations avec leurs cardinalités, on obtient le modèle conceptuel des données dans le formalisme entité-association

### 3.2 - Passage au modèle relationnel

Il existe des algorithmes de passage du modèle entité-association au modèle relationnel, c'est-à-dire du MCD au MLD. Nous ne donnerons ici que les algorithmes principaux sous la forme de trois règles utilitaires. :

Règle 1 : Les entités du MCD deviennent les tables du MLD ; les identifiants deviennent des clés ; les propriétés deviennent des attributs.

Règle 2 : Pour une association avec une cardinalité 1,1, on ajoute l'identifiant cible (de l'entité lointaine) à la table source (entité proche) et l'association est supprimée.

Règle 3 : une association dont toutes les pattes sont de la forme x,n devient une table à part entière dont l'identifiant est construit par concaténation des identifiants des entités adjacentes ; les propriétés de l'association deviennent des attributs de cette nouvelle table.

Exemple : Appliquons ces trois règles à l'exemple de la maison de négoce en vin. La figure donnée plus haut exprimant le MCD, la règle 1 permet de définir trois tables :

VIN (numvin, type, mil)

CLIENT (numclient, nomclient, prenomclient, adresse)

DATE (date)

La règle 2 est sans objet ; la règle 3 permet d'obtenir une quatrième table :

ACHAT (numvin, numclient, date, nbout)

La clé composée numvin, numclient, date peut être d'un emploi peu commode, quoique rigoureux ; pour des raisons de simplification, on peut être amené à la remplacer par une clé simple numachat équivalente, avec les dépendances fonctionnelles numachat à numvin, numachat à numclient, numachat à date, ce qui conduit à la table transformée :

ACHAT (numachat, nbout, #date, #numvin, #numclient)

Dans ce cas, la table DATE n'est d'aucun intérêt et on peut supprimer sans dommage cette table, qui, isolée, n'est d'aucune utilité ; on aura finalement la modèle logique de données exprimé par les trois tables suivantes :

VIN (numvin, type, mil)

CLIENT (numclient, nomclient, prenomclient, adresse)

ACHAT (numachat, nbout, date, #numvin, #numclient)

On verra au chapitre 6 l'application des trois règles précédentes à d'autres exemples.

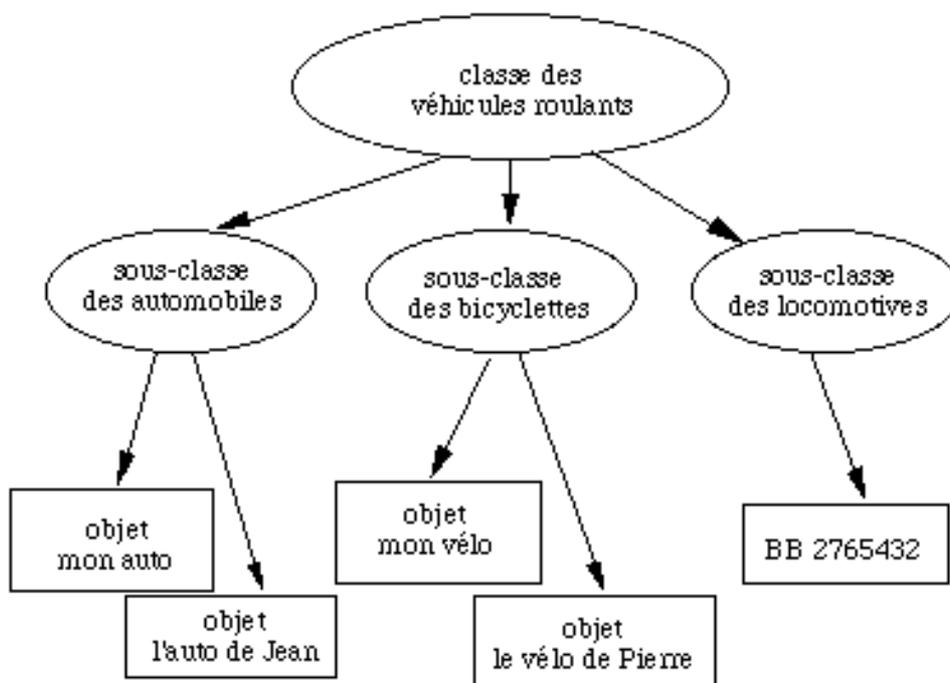


## 4 - Modèle orienté objet



Les bases de données objet sont apparues vers la fin des années 80. Les techniques objet reposent sur les trois concepts fondamentaux d'objet, de classe et d'héritage. Un objet est une entité caractérisée par un aspect statique appelé état ou consistant en propriétés et un aspect dynamique consistant en méthodes qui définit le comportement de l'objet à des événements. Parmi les propriétés, il en est une qui définit sans ambiguïté l'objet : l'identifiant. Les objets de même type appartiennent à une classe qui est donc caractérisée par des propriétés et des méthodes communes à plusieurs objets. Les méthodes sur les objets sont activées par des requêtes sous forme de messages. Une classe peut contenir des sous-classes qui héritent des caractéristiques de la classe.

Exemple : La classe des véhicules roulants contient des sous-classes : celle des automobiles, celle des bicyclettes, celle des locomotives (figure ci-dessous). Les propriétés de la classe des véhicules roulants sont le nombre de roues, la vitesse maximum. Par héritage, ces propriétés sont aussi celles des automobiles, des bicyclettes et des locomotives. De même la méthode "rouler" de la classe des véhicules roulants est aussi méthode pour les sous-classes automobiles, bicyclettes, locomotives.



Paradoxalement la notion d'objet est contradictoire avec le principe de séparation des données et des traitements. Données et traitements correspondent ici aux notions de propriétés et de méthodes. Or, ces deux caractéristiques sont inséparables dans un objet (encapsulation). Il s'agit donc d'une analyse bien différente de celle qui a conduit aux bases de données "traditionnelles" y

compris les bases de données relationnelles. Les bases de données objet qui sont des organisations d'objets sont apparues pour pallier les insuffisances du modèle relationnel. Ce dernier, en effet, n'est pas adapté à la gestion des entités complexes : longs textes, images, plages sonores,... qui ne peuvent pas toujours être incorporés simplement dans des "champs" d'une table. Les bases de données objet pourraient apporter une réponse à ce problème, mais la technologie objet est encore jeune, non totalement normalisée et non totalement stabilisée. De fait les bases de données objet appartiennent actuellement à deux catégories :

- les bases de données objet issues des modèles relationnels et proposées par les éditeurs classiques : Oracle, Informix, Sybase, ... Il s'agit d'extension des systèmes relationnels et en particulier du langage de requête SQL.
- Les bases de données véritablement objet qui ne reprennent pas le modèle relationnel et qui sont proposées par des jeunes sociétés comme O2 Technology, Gemstone,...



## 5 - Produits à tendance multimédia



Le panorama des produits proposés pour la gestion des données multimédias est à la fois vaste et varié. On peut toutefois les classer en quatre catégories :

- La première catégorie correspond à des systèmes dédiés, développés pour un besoin particulier et ayant nécessité une analyse très profonde et donc un coût d'investissement important. Il s'agit de produits performants mais restreints à une utilisation particulière. En fait, ce n'est pas le produit qui est vendu mais le savoir-analyser et le savoir-réaliser de la société qui le propose. Un exemple en est fourni par le système Libra de consultation numérique de la Bibliothèque Nationale de France., développé par la société canadienne Alex Informatique.
- La seconde catégorie correspond aux bases de données orientées objet. Ces bases sont commercialisées mais sont encore loin de dominer le marché. Par ailleurs, elles nécessitent de la part de l'utilisateur une formation poussée, notamment en matière d'analyse. Ces systèmes de gestion d'objets ont été définis dans un article célèbre (The Object-Oriented Database System Manifesto) ; ils peuvent être définis comme une structure cohérente d'objets permanents (on dit plus techniquement "persistants") partagés entre plusieurs utilisateurs.
- La troisième catégorie est celle des SGBD relationnels avec "fonctionnalités" multimédias, que l'on désigne souvent sous le nom de SGBDR étendus. Tous les grands éditeurs de SGBD relationnels proposent des produits de ce type : Oracle, Sybase, Informix,...
- La quatrième catégorie est celle des SGBD traditionnels, le plus souvent utilisables sur des systèmes micro-informatiques en réseau. Le multimédia y est traité de façon simple : les fichiers images, son, vidéo, voire texte, sont repérés par leur adresse de stockage dans les champs des tables. La récupération de ces adresses permet de les retrouver et de les afficher/entendre/visualiser moyennant une interface ad hoc (programmes spécifiques) que l'utilisateur peut réaliser lui-même (nous en verrons quelques exemples plus loin), ou l'usage d'assistants (modules pré-programmés permettant l'accès agréable à ces médias)..

Le progrès technologique aidant et les utilisations de données multimédias progressant, il est clair que ce panorama est extrêmement changeant et il faudra attendre encore un peu pour voir se dégager, dans la gestion d'objets multimédias, des standards stables et donc rassurants.

## 6 - Bibliographie



A. FLORY, F. LAFOREST	Les bases de données relationnelles	Economica
G. GARDARIN	Bases de données	Eyrolles
S. REYMANN	Informatique DECF, MSTCF (*)	Foucher
S. MIRANDA, J-M. BUSTA	L'art des bases de données ; Introduction aux bases de données	Eyrolles
S. MIRANDA, J-M. BUSTA	L'art des bases de données ; Les bases de données relationnelles	Eyrolles
C.J. DATE	An Introduction to Database Systems	Addison-Wesley
G. GARDARIN, P. VALDURIEZ	SGBD avancés	Eyrolles
G. GARDARIN	Internet/Intranet et bases de données	Eyrolles
A. MESGUICH, B. NORMIER	Comprendre les bases de données( $\mu$ )	Masson
G. GARDARIN, P. VALDURIEZ	Bases de données relationnelles ; analyse et comparaison des systèmes	Eyrolles
G. GARDARIN	Bases de données ; objet et relationnel	Eyrolles
D. NANJI, B. ESPI NASSE, B. COHEN, H. HECKENROTH	Ingénierie des systèmes d'information avec Merise ; vers une deuxième génération	Sybex
N. KETTANI, D. MIGNET, P. PARE, C. ROSENTHAL-SABROUX	De Merise à UML	Eyrolles

(\*) pour débutants



# Gestion de l'information multimédia

## Technologies

---

Objectifs

Introduction

1. Technologies d'acquisition
2. Technologies de stockage et d'archivage
3. Technologies d'exploitation et de diffusion
4. Bibliographie

 Téléchargement du chapitre (chap5.pdf - ... Ko)

---

Concepteurs : 

Gérard-Michel Cochard

# Introduction

---



La construction des systèmes d'information multimédias s'appuie sur un ensemble de technologies dont la plupart ont vu le jour que très récemment. Elles concernent les différentes phases de la construction : acquisition des données multimédias, stockage et archivage, exploitation et diffusion.

---

⇒ Sections du chapitre : [1](#) | [2](#) | [3](#) | [4](#)

# 1 - Technologies d'acquisition



Les technologies d'acquisition consistent essentiellement à numériser les données et documents multimédias. Elles sont dédiées à chaque type de média et correspondent à une activité très importante puisque l'on considère qu'environ 95% des documents ne sont pas numérisés. De fait, le marché des outils de numérisation est très florissant.

## Les textes

Les textes peuvent être numérisés de deux façons. La méthode la plus simple consiste à dactylographier ou re-dactylographier le texte en utilisant un logiciel de traitement de texte ou des outils logiciels voisins comme les tableurs. Ces logiciels effectuent un codage informatique des caractères d'imprimerie et signes constituant les textes.

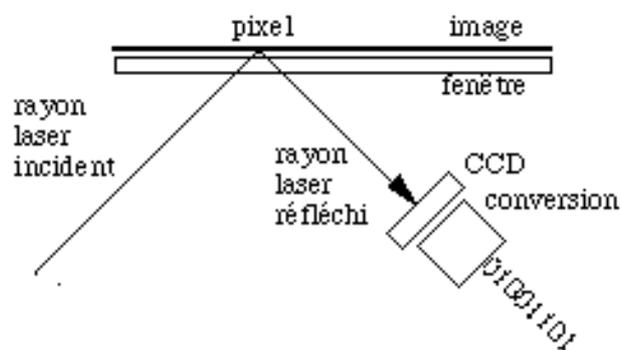
Une autre façon d'acquérir les données informatiques représentant un texte est de considérer ce texte comme une image et de lui appliquer les techniques de numérisation des images. Bien entendu, on ne pourra plus par la suite considérer ce texte-image comme un texte modifiable symbole par symbole. Pour cela, il faut lui appliquer un traitement à l'aide d'un logiciel OCR (Optical Character Recognition) qui effectue une reconnaissance des formes contenues dans le texte-image et les traduit en termes de caractères d'imprimerie. Le texte-image est ainsi balayé et analysé et cette opération, suivant les divers logiciels OCR du marché, peut être plus ou moins longue et le résultat plus ou moins performant.

La manipulation du texte numérisé entre diverses applications nécessite une normalisation du format de codage. Le code universellement employé est le code ASCII (American Standard Code for Information Interchange) qui représente un caractère sur 7 bits. Les ordinateurs traitant des blocs dont la taille est multiple de mots de 8 bits (octets), on peut ajouter un "0" devant les 7 bits du code ASCII pour obtenir des octets représentant des caractères. Toutefois, le code ASCII ne permet, avec 7 bits, que le codage de 128 caractères distincts ce qui est peu. Pour augmenter les possibilités, on peut étendre le code ASCII à 256 caractères supplémentaires en ajoutant un "1" devant les 7 bits (second jeu de caractères). Malheureusement, ce second jeu de caractères n'est pas normalisé et un caractère numérique peut avoir des représentations différentes suivant les plates-formes utilisées. Par ailleurs, il existe aussi des formats "constructeurs" comme par exemple, RTF pour les traitements de textes et SYLK pour les tableurs. Des logiciels de conversion doivent alors être utilisés ; il en existe beaucoup et sont, soit spécialisés à des tâches de conversion uniquement, soit partie intégrante des traitements de texte.

## Les images

La plupart des images existent déjà sur papier et la numérisation de documents papier à l'aide d'un scanner est maintenant une opération banale. Un scanner utilise un procédé optique pour numériser un document : un faisceau laser est envoyé sur un point du document et se réfléchit avec une composition lumineuse qui dépend de la couleur du point ; ce faisceau réfléchi est analysé par une barette de cellules CCD (Charged Couple Device) qui sont des capteurs photosensibles produisant une valeur numérique suivant l'intensité du faisceau lumineux reçu (figure ci-dessous).

L'opération est effectuée point par point et ligne par ligne par balayage régulier du document. Le procédé revient donc à capter la couleur des points situés sur une grille matricielle. La maille de cette grille, c'est-à-dire la distance entre deux points voisins est définie à partir de la résolution réglable du scanner et exprimée en points par pouce (dpi : dots per inch). Les résolutions usuelles vont de 72 dpi à 1200 dpi ce qui correspond à des distances interpixels de 0,3 mm à 0,021 mm respectivement. Certains scanners améliorent la qualité de l'image numérique obtenue en calculant par interpolation la couleur de points intermédiaires. Le choix de la résolution est guidé par deux caractéristiques antagonistes, la qualité et la taille qui augmentent toutes deux avec la résolution. Une photo d'identité de 4cm sur 5cm en 16 millions de couleurs et numérisée avec une résolution de 600 dpi conduit à un fichier brut dont la taille est 3,3 Mo alors que la même photographie en 256 couleurs et numérisée avec une résolution de 72 dpi conduit à une taille de fichier brut de 16 Ko.

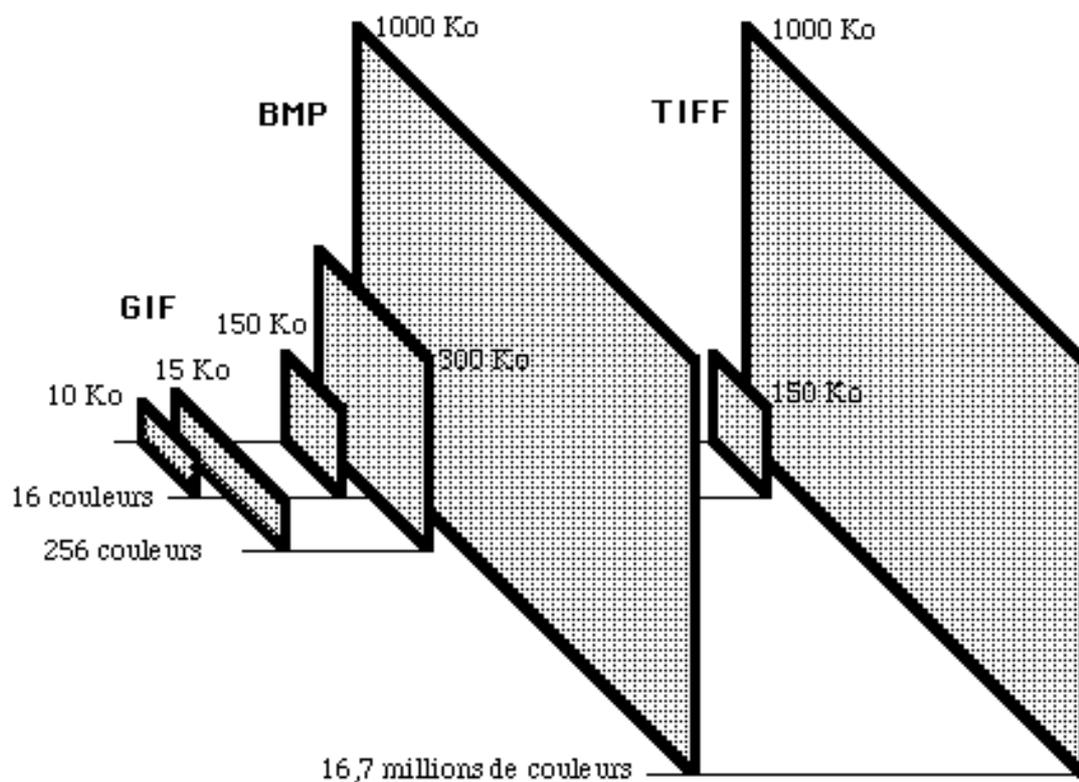


La seconde possibilité d'obtention des images numériques est leur production directe. Elle peut s'effectuer par dessin de l'image à l'aide d'un logiciel spécifique. Pour les photographies, les appareils de photographie numérique apparus récemment sur le marché grand public, produisent des fichiers enregistrés sur disquettes ou sur cartes au format PC

Une dernière méthode d'acquisition numérique d'images est l'utilisation du procédé Kodak qui permet l'obtention sur un CD-ROM particulier appelé CD-Photo de transformer des négatifs ou des diapositives classiques de photographies en un enregistrement sur disque des mêmes photos, mais numérisées.

Les formats d'image sont assez divers et aucun ne s'est encore véritablement imposé pour des applications multimédias. Les formats se partagent entre des formats "constructeurs" et des formats issus d'organismes de normalisation.

Le format BMP est le format d'image matricielle de Windows ; le format PICT est celui utilisé sur les plates-formes Macintosh. Le format TIFF est un format d'échange utilisable sur plusieurs plates-formes. Le format GIF de Comuserve est un format intéressant du fait du faible encombrement résultant des méthodes de compression utilisées ; cette compression est sans perte et le nombre de couleurs va jusqu'à 256 . La comparaison de ces trois formats est illustrée ci-dessous .



Le format JPEG est un format standardisé impliquant une compression avec perte et la possibilité d'avoir 16,7 millions de couleurs. Les formats JPEG et GIF sont actuellement les seuls à être interprétés par les navigateurs du Web ce qui les avantage indiscutablement.

## Les sons

La numérisation des sons s'opère par des cartes spécifiques à partir de sons enregistrés sur des bandes classiques analogiques. Les cartes Sound Blaster en sont la référence principale. Des outils de retouche du son sont également fournis avec ces cartes .Le son numérisé est obtenu suivant divers formats :

- PCM (Pulse Code Modulation),
- MPCM (Delta PCM) où le codage porte sur les différences entre les valeurs successives échantillonnées,
- ADPCM (Adaptive Differential PCM) où des interpolations son effectuées afin de diminuer le volume.

## La vidéo

L'acquisition de la vidéo numérique fait appel également à toute une famille de cartes spéciales aux fonctions diverses. Les cartes de numérisation effectuent la simple conversion du son analogique en son numérique par les techniques d'échantillonnage, de quantification et de codage. Elles peuvent également effectuer la compression nécessaire étant donné le volume de données binaires obtenues par numérisation. Elles permettent d'obtenir un son numérique sous divers formats : MPEG, AVI, MOV, etc... selon les utilisations qui en seront faites.

## Vers un format unique ?

Les formats des données numériques issues de la digitalisation des différents médias sont divers et on désigne par Twain les dispositifs qui permettent une conversion dans un format d'échange.

Toutefois, la recherche d'un format spécifique au multimédia a conduit à des formats adaptés :

- MHEG (Multimedia and Hypermedia information coding Expert Group) propose une structuration des données multimédias basée sur l'utilisation du concept d'objet ;
- MPEG-7 est une norme en préparation dédiée au multimédia et proposant une description des contenus ;
- HTML 4 est un standard de description de document Web intégrant des traitements en java ou javascript et permettant un affichage dynamique ;
- HyTime (Hypermedia Time bases Structuring Language) propose une extension de SGML (Standard Generalized Mark up Language) adapté à l'établissement de liens et à la synchronisation des documents multimédias.



## 2 - Technologies de stockage et d'archivage



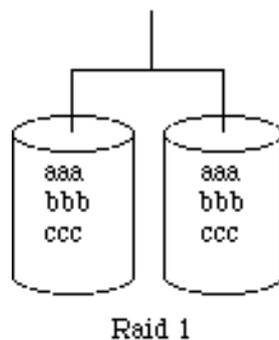
Les bases de données multimédias répertorient des données numériques extrêmement volumineuses que les méthodes de compression ne suffisent pas à amoindrir. Le besoin de décompresser ces données pour leur lecture est d'ailleurs à prendre en considération.

Les supports nécessaires actuels peuvent être classés en disques et bandes et les technologies associées sont de type magnétique, optique ou magnéto-optique. Les critères de choix ou de comparaison sont au nombre de quatre : capacité de stockage (quantité d'information enregistrable), temps d'accès aux données (critique pour une utilisation en temps réel), coût (facteur inévitable) et réinscriptibilité (possibilité d'écritures successives).

### Les disques magnétiques

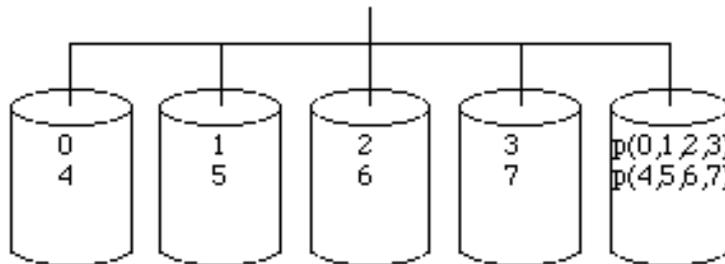
Les disques magnétiques sont utilisés depuis longtemps en informatique classique. Le disque "dur" de l'ordinateur reste encore le support de mémorisation le plus simple et le plus confortable. Les capacités des disques magnétiques utilisant la technologie Winchester atteignent maintenant des valeurs importantes mais encore insuffisantes pour le multimédia. Les disques durs des micro-ordinateurs usuels peuvent être de l'ordre de 4 Go ce qui, compte tenu des applications, laisse une place encore trop réduite aux données et ne permettent pas l'implémentation des données multimédias. Il faut donc adjoindre des batteries de disques supplémentaires en veillant à des temps d'accès satisfaisants ou choisir des configurations spéciales incorporant des disques à haute capacité (jusqu'à 23 Go) et à temps d'accès réduit (jusqu'à 20 ms). Récemment, les disques RAID (Redundant Array of Inexpensive Disks) sont apparus et fournissent des solutions intéressantes. Les objectifs recherchés de ces disques, en dehors de la capacité, sont un haut débit obtenu par une lecture parallèle de plusieurs disques simultanément et une tolérance aux pannes obtenue simplement par redondance. Il existe plusieurs catégories de disques RAID :

- RAID 0 : les données sont réparties sur plusieurs disques
- RAID 1 : un disque est apparié à un disque miroir



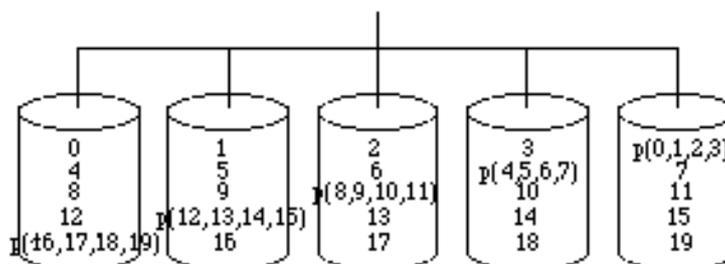
Raid 1

- RAID 3 : les données sont réparties par octets sur plusieurs disques et un disque supplémentaire enregistre la parité (figure ci-dessous) :



Raid 3

- RAID 5 : les données sont réparties par blocs sur plusieurs disques et la parité est également répartie entre ces disques :

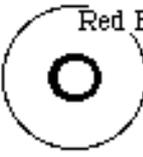
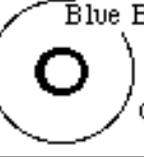


Raid 5

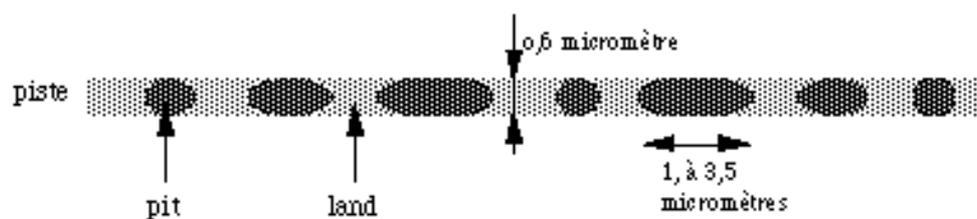
- RAID 6 : plusieurs paires des disques RAID 1

## Les CD

La famille des CD (Compact Disk) est née avec le multimédia et contribue à sa diffusion. Elle comprend des disques en général non réinscriptibles contrairement aux disques magnétiques et correspond à une utilisation en consultation. La technologie est basée sur les propriétés optiques de la matière. La famille des CD comprend actuellement les CD-Audio, les CD-ROM, les CD-I, les CD-ROM/XA, les CD-Photo, les CD-Vidéo, les CD-Extra. Chaque type de CD obéit à un standard précis désigné par Red Book, Yellow Book, etc,...

 <p>Red Book CD-Audio</p>	 <p>Yellow Book CD-ROM CD-ROM/XA</p>	 <p>Green Book CD-I</p>
 <p>Orange Book CD-Photo CD-R</p>	 <p>White Book CD-Vidéo</p>	 <p>Blue Book CD-Extra</p>

Extérieurement, un CD est un disque en matière plastique de 12 cm de diamètre recouvert d'une couche réfléchissante ; intérieurement, il comporte plusieurs couches servant à l'enregistrement des données. Les informations binaires sont représentées par des "pits", cuvettes microscopiques, et des méplats séparateurs de pits appelés "lands". Les lands et les pits représentent des "0", le passage d'un land à un pit ou d'un pit à un land mémorise un "1" (voir ci-dessous). La lumière d'un faisceau laser incident provoque sur un pit une dispersion lumineuse tandis que sur un land on a une simple réflexion. Contrairement aux disques durs, la vitesse linéaire est constante (à partir de 200 tours par minute). Un CD est divisé en "frames" et secteurs (comportant 98 frames).



Les standards d'enregistrement comprennent le standard High Sierra, mis au point par plusieurs constructeurs, le standard ISO 9660, adapté aux plates-formes Windows, qui est une amélioration du standard précédent, le standard HFS adapté aux plates-formes Macintosh, le standard Hybride, combinaison des deux précédents, le standard Rock Ridge adapté aux plates formes UNIX. Le taux de transfert dépend du lecteur de CD et est supérieur à 150 Ko/s.

Le CD-Audio, bien connu des amateurs de musique, a remplacé le disque vinyle à partir de 1982 et est adapté à l'enregistrement sonore. Son standard est connu sous le nom de Red Book. Il permet l'enregistrement de 74 minutes de son non compressé (codage sur 16 bits et fréquence d'échantillonnage de 44,1 Kz).

Le CD-ROM, apparu en 1983 est une généralisation du CD-Audio aux données informatiques. Sa capacité de 640 Mo correspond à 460 disquettes de 1,4 Mo. Son standard est connu sous le nom de Yellow Book et comprend deux modes d'enregistrement, le mode 1 pour les données informatiques et le mode 2 pour les données audio et vidéo compressées.

Le CD-I est lié à l'histoire du Multimédia. Mis au point par Philips et Sony, ce CD était destiné au grand public : utilisation sans ordinateur avec un lecteur dédié et visualisation avec le récepteur TV. Son standard est connu sous le nom de Green Book. Mais il a été détrôné par le CD-ROM, son principal concurrent. Ce type de CD semble maintenant abandonné.

Le CD-ROM/XA est une extension du CD-ROM opérant un compromis entre les standards Yellow Book (avec compression ADPCM du son) et Red Book (entrelacement de secteurs)

Les CD précédents sont des CD préenregistrés destinés uniquement à une lecture. Ils ne sont pas réinscriptibles. Le CD-R (Recordable), qui obéit au standard Orange Book, et qui a été mis au point en 1990 par Philips et Sony est un CD que l'on peut enregistrer soi-même avec un graveur de CD. Il est quelquefois désigné sous le nom de CD-WORM (Write Once Read Many). Après enregistrement, il se comporte comme un CD-ROM. A noter qu'il est possible de l'enregistrer en plusieurs fois (mode multisession).

Il existe toutefois des CD réinscriptibles appelés PC/CD. Peu répandus, ils correspondent à plusieurs fois le coût d'un CD-R.

Le CD-Photo, inventé par Kodak en 1992 est une sorte de CD enregistrable permettant le stockage d'images (jusqu'à 800) ou de son (jusqu'à 72 minutes) ou un mélange des deux. Il est proposé sous plusieurs formules : CD-Photo Pro, CD-Photo Master, CD-Photo Portfolio, Catalog Disk.

La vidéo possède, elle-aussi, son standard de CD (White Book), correspondant au CD-Vidéo qui comporte des pistes audio et vidéo.

Enfin, le CD-Extra, obéissant au standard Blue Book, est destiné à remplacer le CD-Audio par amélioration de ce dernier

Les CD constituent un support de diffusion grand public très apprécié du fait de ses qualités relatives à son prix, à sa fiabilité, à ses standards reconnus, à sa capacité de stockage et à son faible encombrement. Le marché du CD a fait naître toute une industrie au plan international

Pour des applications nécessitant une durée de vidéo plus importante, le CD est cependant assez limité. Il sera vraisemblablement remplacé à terme par le DVD (initialement Digital Vidéo Disk, puis Digital Versatile Disk), dont la capacité de stockage permet l'enregistrement de plus de 2 h de vidéo numérique. Destiné initialement à remplacer la cassette VHS, il est en passe de remplacer les membres de la famille des CD que nous avons énumérés précédemment. Le seul obstacle à son développement est le parc actuel de micro-ordinateurs, équipés de lecteur de CD-ROM et non de lecteur de DVD ; mais les nouveaux matériels seront équipés de lecteurs universels qui peu à peu favoriseront l'essor du DVD. Son intérêt est indéniable puisqu'il correspond à une capacité de stockage équivalente à celle de 12 CD-ROM et à un débit allant jusqu'à 10 Mbits/s. La technologie du DVD s'appuie sur le doublement des couches inscriptibles et sur l'augmentation des "creux" et des "bosses" qui matérialisent les symboles binaires.

## Les bandes

Les bandes magnétiques sont des supports de type linéaire/séquentiel contrairement aux disques qui permettent un accès direct aux données. Elles ne se prêtent donc pas à une consultation sélective, mais plutôt à l'archivage pour des raisons de conservation ou de sécurité. Néanmoins, il est possible de les utiliser pour le déroulement de séquences sonores ou vidéo. L'enregistrement sur bande peut s'opérer sur des pistes linéaires ; la capacité de ces bandes peut atteindre une quarantaine de gigaoctets et le débit une dizaine de mégaoctets par seconde. Il peut également s'opérer de manière "hélicoïdale" où les pistes sont diagonales par rapport à l'axe de la bande. ; dans ce cas, la capacité peut atteindre une centaine de gigaoctets et le débit environ 150 mégaoctets par seconde.



## 3 - Technologies d'exploitation et de diffusion

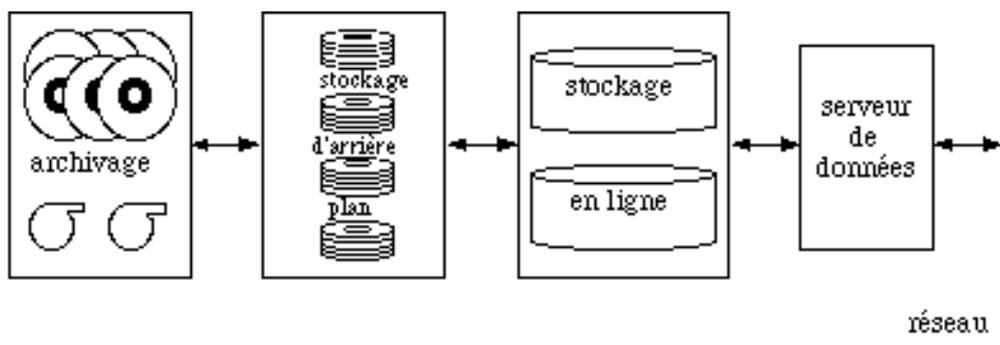


Les données multimédias sont généralement disponibles à partir de la lecture sur une station multimédia d'un CD-ROM les contenant, soit à partir d'un poste client relié à un serveur par un réseau informatique. Le réseau peut être local ou distant. Dans le cas d'un réseau local, plusieurs architectures sont disponibles auprès de constructeurs spécialisés. Elles sont toujours étudiées de manière spécifique pour répondre à un besoin très précis. Dans le cas d'une diffusion ou d'une exploitation à grande distance, s'impose inévitablement Internet et sa composante multimédia que constitue le World Wide Web.

### Mode local

Une architecture typique en réseau local comprendra un réseau à très haut débit, ce qui est réalisable pour des distances peu importantes par des liaisons dont les débits vont de 10 à 100 Mbits/s. Le serveur "média" possède les fonctionnalités de gestion de la base de données, de gestion des requêtes des utilisateurs. Le dispositif de stockage peut avoir trois composantes :

- le stockage en ligne basé sur l'utilisation de disques magnétiques
- le stockage d'arrière plan basé sur l'utilisation de CD ou DVD disposés dans un juke box
- l'archivage pour copie et sécurité, non nécessairement en ligne avec le serveur (CD, bandes).



Le stockage d'arrière plan, qui fait appel à des organes mécaniques, n'est pas aussi performant que le stockage sur disques durs en ce qui concerne les temps d'accès. Néanmoins, la technologie des juke boxes a fait d'importants progrès ces dernières années et il est maintenant courant de trouver sur le marché des juke boxes gérant 300 disques CD ou DVD avec 2 ou 4 lecteurs pour un coût inférieur à 150 KF.

La diffusion locale utilise des infrastructures de réseau haut débit (ou large bande). Elle peut être basée sur une architecture de réseau local du commerce : Novell Netware ou Windows NT ; dans

cette éventualité, des développements spécifiques doivent être réalisés, au niveau serveur comme au niveau client.

Les logiciels auteurs que constituent des produits actuels comme Director, Authorware, Toolbook, Apple Media Tool, ... permettent la réalisation d'interfaces de qualité. Certains produits de gestion de base de données peuvent proposer des fonctionnalités presque équivalente (4D sur Macintosh, Access sur PC, ...). Dans tous ces cas, l'intervention de professionnels confirmés est requise si l'on souhaite, dans un délai raisonnable, parvenir à une réalisation satisfaisante.

Il existe maintenant une autre possibilité qui est fournie par la construction d'un Intranet que l'on peut définir rapidement comme l'utilisation d'Internet à des fins particulières et pour un groupe d'utilisateurs définis. Cette construction n'est pas fondamentalement différente de celle d'un service de consultation à distance basée sur l'utilisation du World Wide Web que nous mentionnons ci-dessous. En effet, que le serveur soit distant ou proche du poste client, les techniques mises en jeu sont les mêmes.

## Mode distant

La diffusion à distance s'appuie actuellement sur le protocole HTTP (HyperText Transfer Protocol) qui est celui du World Wide Web sur Internet. Dans ce cas, les documents multimédias sont distribués par un serveur Web qui doit être en communication avec la base de données ce qui peut s'opérer de plusieurs manières, soit par programmation spécifique (incluant éventuellement des requêtes SQL) en utilisant CGI (Common Gateway Interface), soit par utilisation de standards comme ODBC (Open Data Base Connectivity) ou équivalents (API, ASP, ...). A ce stade de l'exposé, il convient de donner quelques indications de base sur ce qu'est Internet, le WWW et le protocole HTTP, le langage HTML, les connecteurs de base de données.

## Internet

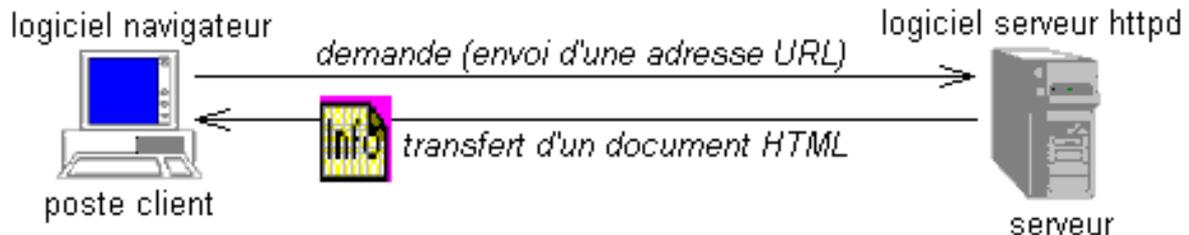
Internet, qui n'est autre qu'une fédération de réseaux existants (un bon exemple de mutualisation !) possède deux caractéristiques principales qui ont fait son succès :

- L'utilisation de la famille de protocoles TCP/IP, simples à mettre en œuvre et figurant dans le noyau du système d'exploitation UNIX qui équipait au départ la grande majorité des machines connectées.
- L'utilisation de liaisons à hauts débits (ou à larges bandes) permettant ainsi le transport, dans des temps raisonnables, de données binaires volumineuses (cas des données de type son ou image).

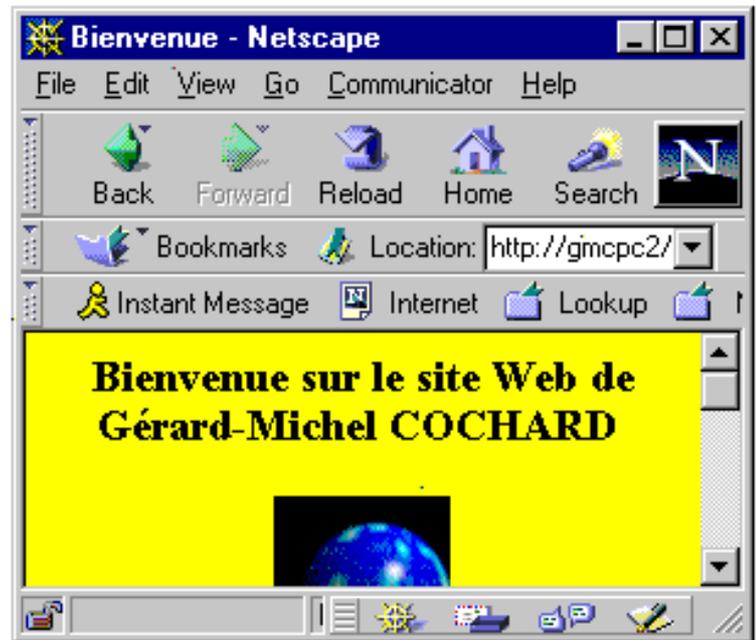
## WWW et HTTP

Le grand essor d'Internet est dû, à partir de 1990, à la mise en œuvre du protocole HTTP

(HyperText Transfer Protocol, 1989, T. Berners-Lee), plus connu sous le nom de Web (toile d'araignée) ou de WWW (World Wide Web : toile d'araignée à l'échelle planétaire). Ce protocole, basé sur le concept client-serveur, permet à un poste client de demander l'envoi (Fig. 4.09), par un serveur WWW d'un document multimédia décrit par un langage spécifique HTML (HyperText Markup Language), lui-même issu d'un langage plus complet appelé SGML (Standard Generalized Markup Language).



Sur le poste client, un logiciel appelé navigateur (ou explorateur, ou navigateur, ou bien d'autres noms plus ou moins poétiques comme "brouteur", ...) interprète le fichier HTML reçu du serveur et en assure la visualisation (avec la possibilité d'entendre les sons grâce aux possibilités multimédias des ordinateurs personnels actuels). Les principaux navigateurs du moment sont Netscape 4 (ou Communicator de Netscape) et Internet Explorer 5 (de Microsoft) :



La demande de document s'effectue très simplement par envoi de l'adresse de celui-ci (URL) ; cette demande s'adresse à un serveur Web équipé d'un logiciel httpd (http daemon), qui n'est autre qu'un logiciel serveur de documents HTML. Les logiciels httpd les plus connus sont donnés dans la figure ci-dessous :

httpd	caractéristiques
Apache	destiné initialement aux machines UNIX ; extension récente au monde Windows (NT et 95)
IIS (Internet Information Server)	le produit professionnel de Microsoft pour Windows NT
NIS (Netscape Interprise Server)	le produit de Netscape pour UNIX et Windows NT
WebSite	de O'Reilly ; très populaire aux USA chez les amateurs ; très bonne documentation
PWS(Personal Web Server)	un petit serveur gratuit de Microsoft ; très pratique pour le développement de sites Webs

On trouvera des renseignements plus complets aux références suivantes disponibles sur le Web évidemment :

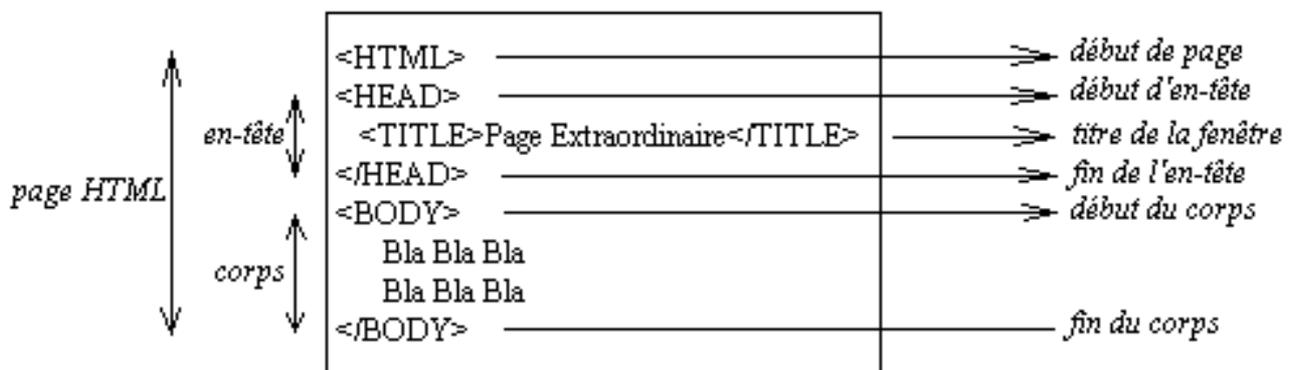
[http://www.yahoo.com/Computers\\_and\\_Internet/Internet/World\\_Wide\\_Web/HTTP/Servers/](http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/HTTP/Servers/)

<http://webcompare.iworld.com/>

## HTML

HTML est un langage de description de document. Il permet de décrire une "page", ensemble séquentiel d'informations comme une page de livre, mais avec une longueur non définie. HTML utilise des marqueurs ou balises (en anglais tags) pour spécifier des mises en forme des éléments du documents.

Une page HTML a la structure décrite dans la figure ci-dessous :

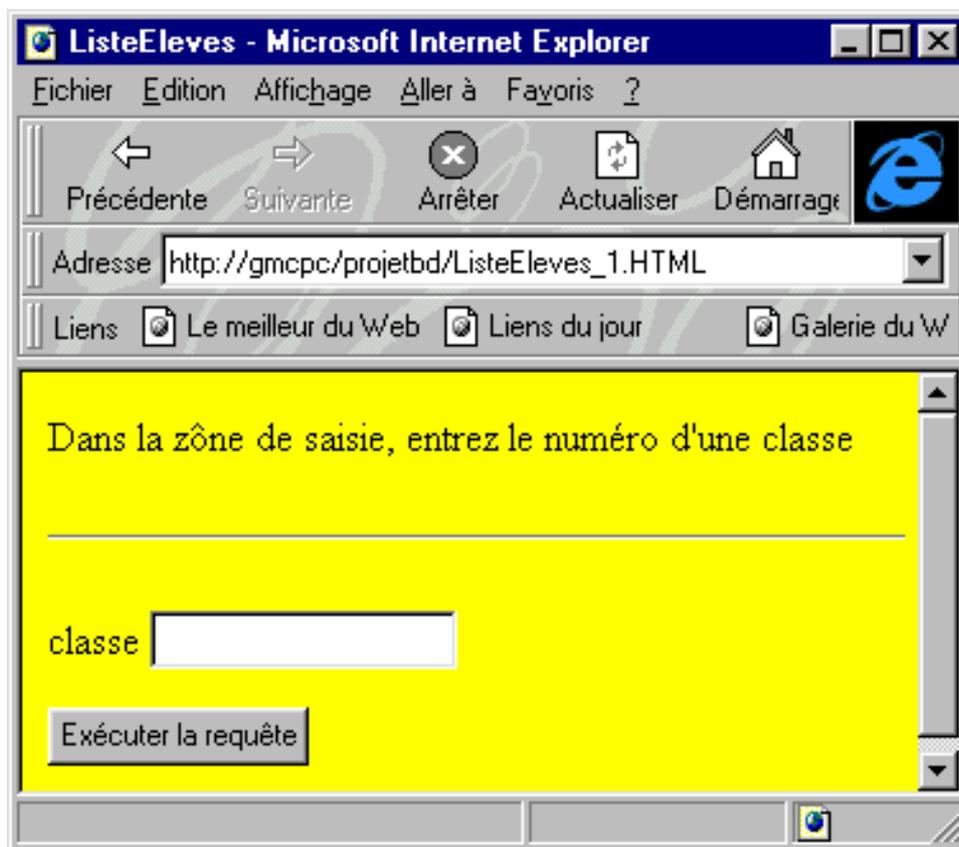


Le document ou page HTML se compose d'une en-tête et d'un corps annoncés chacun par des balises repérables par les signes "<" et ">". Ce qui est affiché dans le document se trouve dans le corps du document. L'en-tête sert à définir un certain nombre de paramètres préalables y compris le titre de la fenêtre qui accueillera le document. Les balises qui permettent de structurer les éléments du documents sont maintenant assez variées (dernières version : HTML 4) et il convient d'avertir le lecteur que ce que l'on "voit" sur l'écran est fortement tributaire du logiciel navigateur utilisé ; il est recommandé d'utiliser toujours la dernière version ce qui n'est pas difficile puisque les deux navigateurs les plus populaires sont entièrement gratuits.

Les améliorations de HTML ont porté sur l'ajout de nombreuses fonctionnalités permettant de réaliser des documents de plus en plus dynamiques. Pour simplifier on peut distinguer trois types d'amélioration :

- L'ajout de balises supplémentaires ; par exemple la balise <LAYER> de HTML 4 permet de superposer des images en distinguant plusieurs couches d'un même document.
- L'incorporation dans HTML de programmes effectuant diverses actions sur le poste client : Java, qui est un langage compilé assez proche du langage C s'est développé ce sens ; il permet de créer des applets, petits programmes compilés qui peuvent être chargés à partir d'une page HTML présente sur le poste client. ; JavaScript, son petit frère, est un langage interprété dont le code figure en clair dans le code HTML, entre les balises <SCRIPT language="JavaScript"> et </SCRIPT>.
- L'utilisation d'un document HTML comme interface de requête permettant à un utilisateur de formuler l'objet de sa recherche a donné lieu à un ensemble de techniques. Tout d'abord la balise <FORM> permet la construction d'un formulaire de requête.

Exemple : Saisie d'un critère de recherche ; la figure ci-dessous montre une fenêtre dans laquelle un document HTML de saisie (formulaire) a été chargé.



Le code HTML associé à ce document est le suivant :

```

<html>
  <head>
    <title>ListeEleves</title>
  </head>

  <body bgcolor="#FFFF00">
    <p>Dans la zone de saisie, entrez le numéro d'une classe </p>
    <br>
    <p><br></p>
    <form action="Scripts/ListeEleves_1.IDC" method="GET">
      <p>classe
      <input type="text" size="20" name="classe">
      </p>
      <p>
      <input type="submit" value="Exécuter la requête">
      </p>
    </form>
  </body>
</html>

```

Ce code nécessite quelques explications et quelques commentaires. Tout d'abord, on repère le formulaire aux balises qui le délimitent : <form> et </form>. La balise <form> indique l'action (attribut "action") qui lui est associée. Ici, le connecteur ODBC, que nous décrivons plus loin est utilisé, mais il y a d'autres façons de faire. La méthode GET est l'un des moyens de retourner des informations vers le serveur ; il n'y a que deux méthodes, GET pour des informations courtes et POST pour des informations plus longues.

La zone de texte est décrite par la balise <input> avec l'attribut type="text" et son nom qui peut être compris comme une variable passée vers le serveur lorsque le bouton "submit", lui-même déclaré par une seconde balise <input>, sera actionné. De fait, lorsque le bouton "Exécuter la requête" sera actionné, la valeur saisie sera communiquée au serveur, par exemple "classe=CM2".

Le serveur httpd analysera cette arrivée de paramètre(s) et actionnera un programme au niveau du serveur susceptible de fournir le service demandé avec le ou les paramètres reçus.

En définitive, l'utilisation de documents HTML est un moyen puissant de constituer des interfaces de requêtes auprès de serveurs de données multimédias ou non. Comme nous le verrons, HTML permet aussi de fournir les réponses aux requêtes via des mécanismes performants dont le seul tort, à notre avis, est d'être encore spécifiques à un matériel donné ou un type de navigateur donné. Mais, ceux qui ont connu la vague informatique des années 80 et la prolifération de systèmes informatiques divers et non compatibles, comprendront qu'une standardisation finira, inévitablement, par s'imposer, généralement selon des critères économiques, après une période de turbulence riche en innovations. Quoi qu'il en soit, HTML est un langage de description qui continue d'évoluer ce qui rend évidemment difficile, dans le moment présent, des développements d'envergure.

## En savoir plus sur HTML

### Hyperliens

Nous n'avons pas encore mentionné une propriété géniale du World Wide Web, utilisée d'ailleurs maintenant dans bon nombre d'applications. Un hyperlien est la possibilité, par un simple clic de la souris, de "naviguer", c'est à dire de passer d'un document à un autre quel que soit la localisation de ce document et ceci de manière totalement transparente vis à vis de l'utilisateur.

Dans sa forme la plus simple un hyperlien n'est autre qu'un mot ou groupe de mots (signalés par une couleur particulière), qui constitue une zone sensible. Un clic de la souris sur cet hyperlien permet de charger un autre document (entre autres variantes) situé sur un serveur quelconque.

Le "surf" sur Internet, que la presse populaire relate souvent, correspond à cet navigation par hyperlien. Cette idée de relier de manière simple les documents les uns aux autres n'est pas neuve, mais le World Wide Web lui a donné toute son importance en portant ses fonctionnalités à l'échelle planétaire.

Qu'y a-t-il derrière un hyperlien ? Tout simplement l'adresse sur Internet d'un document. Une façon simple de procéder est d'utiliser une balise "ancree" <A> avec un attribut "href" indiquant l'adresse du document à charger en cas de clic de la souris.

Exemple : A titre d'illustration, les lignes ci-dessous présentent une portion de document HTML comportant un hyperlien :

```
<A href="http://truc.com/machin.html">Cliquez ici </A>
```

Lorsque l'utilisateur clique sur le groupe de mot "Cliquez ici", le document machin.html situé sur le site Web truc.com sera chargé. Que peut-on inventer de plus simple ?

### Connecteurs base de données

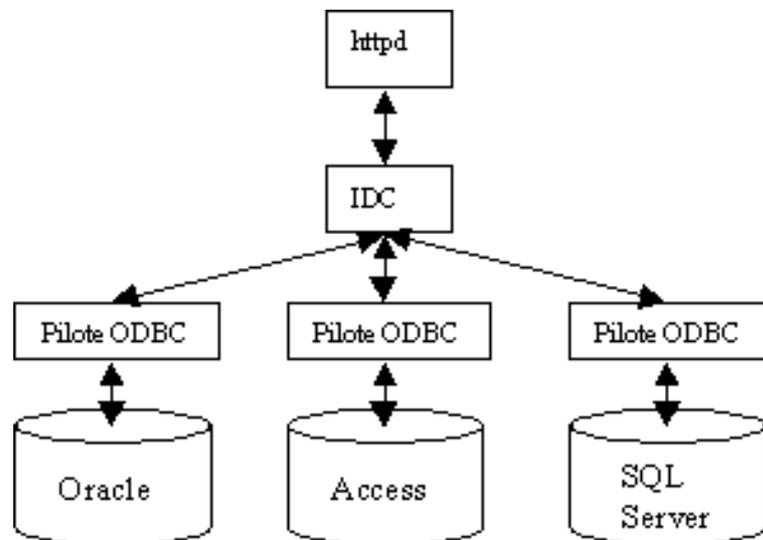
A partir d'un formulaire inclus dans une page HTML, il est possible, comme nous venons de le voir, de transmettre des requêtes à un serveur WWW. Que se passe-t-il ensuite ? Il y a évidemment plusieurs possibilités.

Historiquement, la première technique a été celle des programmes suivant le standard CGI (Common Gateway Interface). Comme la plupart des serveurs, à l'époque de la naissance du Web, étaient implantés sur des machines UNIX, il a été défini des programmes de réponse aux requêtes qui étaient basés sur le système d'exploitation UNIX. Ces programmes pouvaient être programmés directement en shell UNIX, ou en C ; par la suite et encore maintenant, le langage Perl est un

langage typique de gestion des requêtes formulées par un formulaire HTML. Il est bien rare de trouver des programmes CGI autres que pour UNIX, exception faite pour WinCGI, tentative courageuse de procurer des outils pour les malheureux qui possèdent des machines sous Windows.

Mais, bien entendu, ces développements n'avaient pas intégré la puissance du géant Microsoft, lequel a produit nombre d'outils permettant d'exploiter les requêtes des formulaires. L'un de ces outils, ISAPI (Internet Server Application Programming Interface), apporte des améliorations par rapport à la technique CGI. Les programmes (ou scripts) CGI sont des exécutables actionnés par le serveur suite à une requête les mentionnant. ISAPI consiste à utiliser des DLL, c'est à dire des programmes qui restent chargés en mémoire dans le même espace d'adressage que celui du serveur. Il en résulte une plus grande rapidité et une meilleure réponse aux requêtes simultanées.

ODBC (Open DataBase Connectivity) est une interface standard (d'origine Microsoft) d'accès à différents moteurs de base de données relationnelles. La communication s'effectue via un pilote (driver) ODBC qui effectue le lien entre le script (appelé script IDC) et le SGBD :



IDC (Internet Database Connector) est une fonctionnalité du serveur httpd qui permet dynamiquement de pouvoir atteindre une base de données. Cette fonctionnalité se concrétise par le "script" httpodbc.dll qui utilise 2 fichiers principaux de dialogue :

- un fichier texte à suffixe ".IDC" qui contient
  - le nom de la source de données (définie aussi pour le pilote ODBC)
  - la requête en SQL (en général une requête paramétrée)
  - le nom du fichier "modèle" de résultat qui est un fichier de type HTML décrit ci-dessous
- un fichier texte à suffixe ".HTX" (Extended HTML) qui sert de modèle pour la sortie des résultats dans une page HTML.

On peut ajouter à ces deux fichiers le fichier HTML qui permet de saisir les données grâce à un formulaire.

Exemple : Supposons que l'on ait une base de données réalisée, par exemple, sous Access et nommée Ecole.mdb ; pour simplifier, nous supposons que cette base de données est réduite à une seule table :

```
Eleve (numeleve, nomeleve, prenomeleve,
      classe)
```

On définit au préalable un lien permettant de trouver la base, située, par exemple, dans le répertoire C:\Scolarité ; ce lien consiste à désigner cet emplacement par un alias que nous appellerons Ecole ; la donnée d'Ecole permettra donc de retrouver le fichier C:\Scolarité\Ecole.mdb. On définit, pour cela, un pilote ODBC Microsoft Access pour la source de données Ecole (alias). La définition du pilote ODBC et son association à la source Ecole est effectuée à partir du panneau de configuration de Windows 95 (choisir ODBC32).

Sous Access, on prépare la requête suivante : ListeEleves (Recherche des noms et prénoms des élèves qui sont dans la classe que l'on indique (paramètre classe)). En SQL, cette requête se traduit par :

```
SELECT Eleve.nomeleve, Eleve.
prenomeleve
FROM Eleve
WHERE (((Eleve.numclasse)=
[classe])) ;
```

Le fichier IDC pour cette requête sera alors construit comme suit (avec un éditeur de texte ordinaire comme WordPad par exemple ou par utilisation de l'Assistant de conversion HTML d'Access) :

Fichier ListeEleves.IDC :

```
Datasource:Ecole
Template:ListeEleves.htx
SQLStatement:SELECT Eleve.nomeleve, Eleve.prenomeleve
+FROM Eleve
+WHERE (((Eleve.numclasse)='%classe%'));

Password:
Username:
```

Quelques remarques sur ces fichiers :

- On note que l'on peut définir des droits d'accès (Username, Password)
- A l'écriture de la requête SQL, le passage à la ligne est défini par un signe " + ".
- Le paramètre saisi (classe) qui est la paramètre de la requête est noté '%classe%'

Le fichier " modèle de sortie " HTX est construit également avec un éditeur de texte ordinaire (ou un éditeur HTML) ou bien produit en même temps que le fichier IDC par l'Assistant d'Access.

Fichier ListeEleves.HTX :

```

<head>
<title>Liste des Eleves</title>
</head>
<body>
<p> Coucou, voila enfin le résultat !</p>
<hr>
Liste des Elèves de la classe
<%idc.classe%><br><br>
<table border="1" cellspacing="0" width="100%">
<THEAD> <tr> <th>nom de l'élève</th> <th>prénom de l'élève</font></
th> </tr>
</THEAD>
<TBODY>
<%beginndetail%>
<tr>
<td><%nomeleve%><br></td>
<td><%prenomeleve%><br></td>
</tr>
<%endddetail%>
</TBODY>
<TFooter>
</TFooter>
</table>
<p><BR>
<%if CurrentRecord EQ 0 %> Pas de réponses à votre requête <%endif%
>
</body>
</html>

```

A la lecture de ces fichiers, on peut faire les remarques suivantes :

- Les sections répétitives se trouvent entre `<%beginDetail%>` et `<%endDetail%>`
- Les paramètres provenant de la saisie et situés dans le fichier IDC sont désignés par `<%idc.nom_du_parametre%>` comme `<%idc.classe%>`
- Les noms de champ sont désignés par `<%nom_du_champ%>` comme `<%nomeleve%>` ou `<%numclasse%>`
- La variable `CurrentRecord` est un compteur qui part de 0 et s'incrémente chaque fois qu'un nouvel enregistrement est trouvé dans la base.
- La structure de contrôle permet une alternative de sortie :

```

<%if condition %>
texte à afficher si la condition
est vraie
[<%else%>
    texte à afficher si la
condition est fausse]
<%endif%>

```

La condition s'exprime sous la forme

valeur1 opérateur valeur2

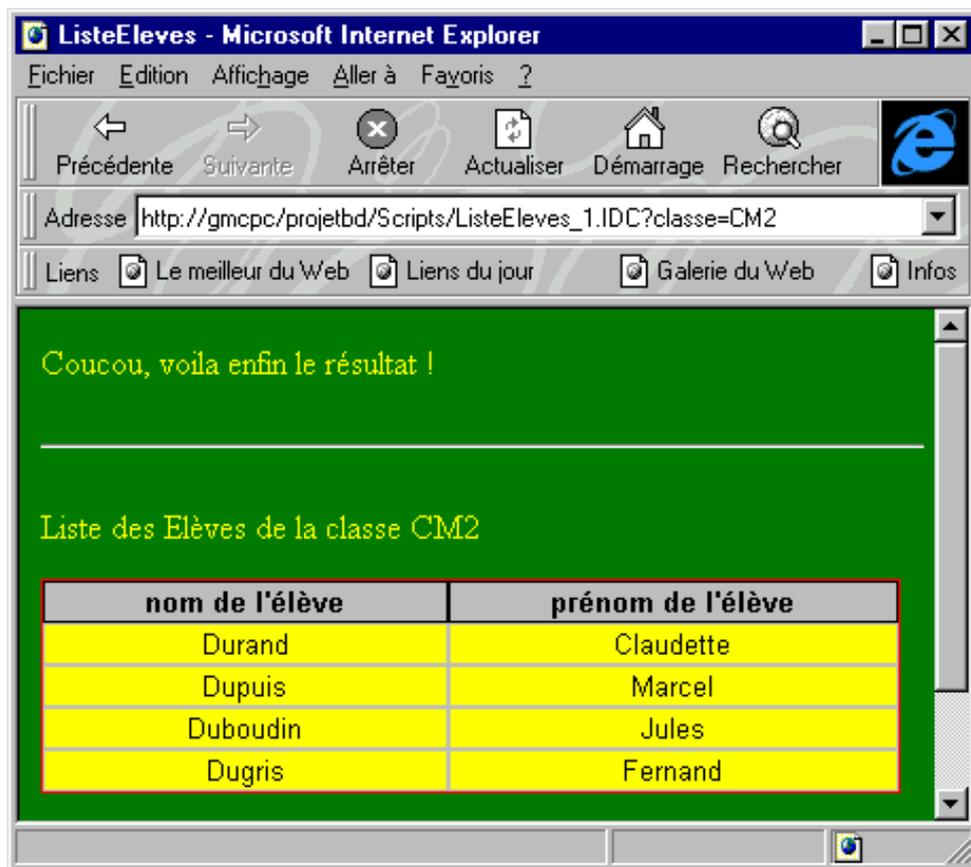
où opérateur se limite à EQ (égal à), LT (inférieur à), GT (supérieur à), CONTAINS (contient).

Les fichiers IDC et HTX sont placés dans le site Web dans un répertoire exécutable Scripts où doit se trouver déjà le script `httpodbc.dll`). Dans notre exemple, le site Web étant dans le répertoire `projetbd`, les fichiers `ListeEleves.IDC`, `ListeEleves.HTX` seront placés avec `httpodbc.dll` dans le répertoire `projetbd/Scripts`.

Il faut également construire (avec un éditeur HTML par exemple) des documents HTML permettant de formuler la recherche (avec des formulaires) . La figure suivante, et le code qui suit, correspondent à cette construction.

On notera dans ce code, le contenu de l'attribut `ACTION` . Les fichiers HTML sont placés dans le répertoire `projetbd` (`projetbd` est aussi le nom du site Web ) et l'adresse du fichier IDC est alors " `Scripts/xxxxx.idc` ".

Le résultat de l'exécution de la requête (fourni par le fichier `ListeEleves.HTX`) est donné plus bas.



La constitution de tous ces fichiers peut paraître fastidieuse. Heureusement, les bases ODBC offrent des fonctionnalités pour construire de manière automatique ces fichiers. Ainsi sous Access, après avoir défini la requêtes ListeEleves, on peut, comme signalé plus haut, les enregistrer au format HTML (menu Fichier de Access 97) ce qui avec l'aide d'un Assistant, construit, dans un format standard les fichiers IDC, HTX, HTML. Il est ensuite possible de retoucher ces fichiers pour les personnaliser, le gros du travail étant fait.



# Gestion de l'information multimédia

## Technologies

Objectifs

Introduction

1. Technologies d'acquisition
2. Technologies de stockage et d'archivage
3. Technologies d'exploitation et de diffusion
4. Bibliographie

 Téléchargement du chapitre (chap5.pdf - ... Ko)

Concepteurs : 

Gérard-Michel Cochard

# Chapitre 5

## Technologies

### Objectifs du chapitre :



- ◆ .....
- ◆ .....
- ◆ .....
- ◆ .....
- ◆ .....



Heure chez vous :

## 4 - Bibliographie



	1. Stockage numérique de masse	
	2. Serveurs "média"	
Etudes :Outils pour la téléconsultation et la distribution sur réseau de documents audiovisuels et multimédias	3. Systèmes de compression	
	4.I. Réseaux et transmission multimédia	INA
	4.II. Réseaux et commerce électronique	
G. DUPOIRIER	Technologie de la GED	Hermes



# Gestion de l'information multimédia

## Systemes d'Information Multimédias

---

### Objectifs

1. Introduction
2. Principes de conception
3. Architecture d'un système d'information multimédia
4. Etudes de cas
5. Bibliographie

 Téléchargement du chapitre (chap6.pdf - ... Ko)

---

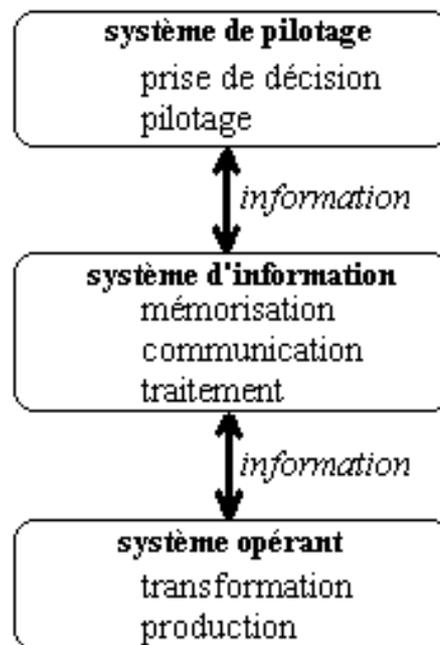
Concepteurs : 

Gérard-Michel Cochard

# 1 - Introduction



Les premières applications informatiques ont principalement porté sur l'automatisation de tâches souvent répétitives dans le cadre des activités d'une entreprise ou d'une collectivité. Elles ont par la suite évolué vers la conception et la réalisation d'outils permettant au gestionnaire de prendre des décisions quant au pilotage de son organisation. C'est vers les années 70 que l'on a été conduit à l'utilisation de bases de données orientées vers une relative indépendance des traitements, puis des infrastructures informatiques. En fait, cette nouvelle orientation s'est accompagnée d'une analyse (ou re-visite) des procédures et des données et a conduit à une tendance à la définition systémique de l'activité des organisations ce qui s'est traduit par l'émergence du concept de système d'information et de méthodologies de conception. L'école française a d'ailleurs été très productive à cette époque et de nos jours, nous utilisons les prolongements de ces recherches et de ces développements, dont la méthode Merise, elle-même à la base d'autres méthodes, constitue un évident et bénéfique résultat.



Le système d'information de l'entreprise est considéré usuellement comme l'intermédiaire entre le système de pilotage ou système décisionnel et le système opérant ou système d'exécution (figure ci-dessus). Son rôle essentiel est un rôle de communication entre ces deux derniers systèmes et également le " milieu extérieur " à l'organisation utilisatrice. Le système d'information gère et maintient les données de l'entreprise, offre aux utilisateurs une possibilité de recherche et d'interrogation. On peut donc résumer aux trois fonctions traitement, communication, mémorisation l'essence d'un système d'information.

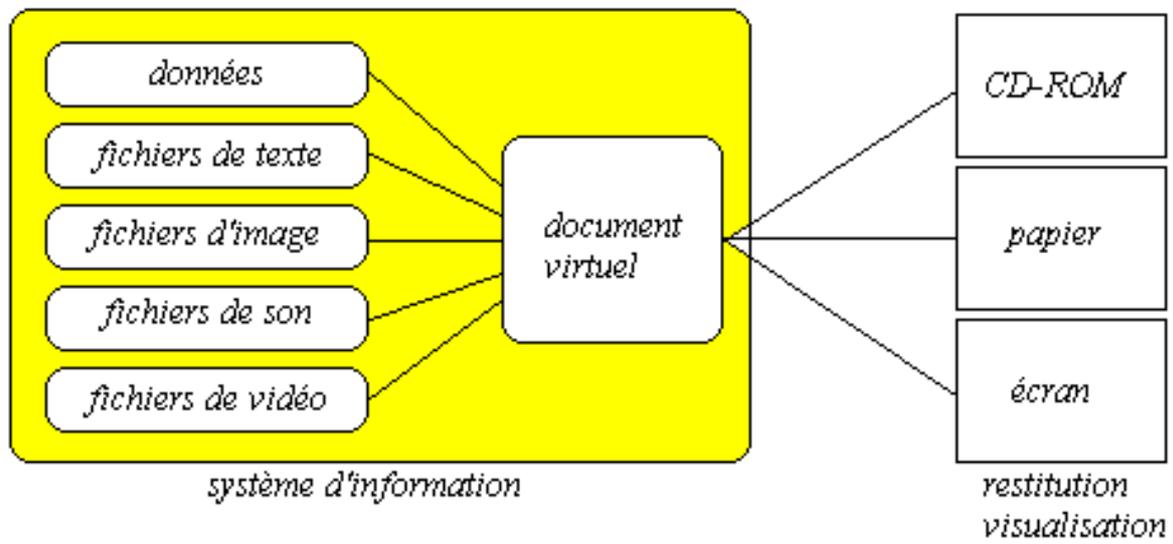
La conception et l'analyse des systèmes d'information impliquent de nos jours l'utilisation d'une méthode rigoureuse dont nous avons déjà donné un aperçu au chapitre sur les bases de données. Sans vouloir s'étendre dans des considérations trop complexes qui nous écarteraient du thème principal de cet ouvrage, nous prenons ici le terme " système d'information " de manière assez générale ; en effet, les systèmes d'information, en fonction de leur finalité, appartiennent à diverses catégories : systèmes de production, systèmes de décision, systèmes documentaires, .... ; nous prendrons donc le terme dans son acceptation globale. Par ailleurs, il convient de définir quelque peu la " matière " traitée, mémorisée et communiquée par le système d'information. Cette matière est définie sous des vocables divers dont les principaux sont les données, les informations, les connaissances, les documents. Il n'existe pas de définition technique précise (et utilitaire), à notre connaissance, de ces concepts, mais nous allons toutefois tenter d'en donner par souci de précision.

Une donnée est une représentation codée d'un fait dans un but de communication ou de traitement et manipulable par un système informatique.

Une information est une perception d'un fait ; elle peut être la signification qu'un utilisateur donne à une donnée ou un ensemble de données ; cette signification peut donc varier d'un individu à l'autre ; elle peut être également une perception d'un fait qui ne se traduit pas (ou pas encore) par des données (une saveur, une odeur sont difficilement informatisables). La valeur d'une variable, l'adresse d'un fichier sont des données ; l'interprétation qu'un individu en donne est une information.

Les données sont régies par des règles qui peuvent être seulement opératoires ou issues d'une expertise plus ou moins complexe. L'ensemble des données et des règles qui les régissent correspondent à la notion de connaissance.

L'apparition des systèmes informatiques permettant la numérisation des textes, des images et des sons a fait naître un nouvel objet : le document virtuel. Le document est un concept au départ lié au support " papier " d'un ensemble de données structurées dans l'objectif de véhiculer des informations ou des connaissances. La numérisation de l'information a étendu très largement cette notion : un document peut être un texte, une image, une plage sonore, ou un mélange de ces différents objets (document composite). Le document possède une intelligibilité beaucoup plus grande qu'une simple donnée , il constitue une forme exhaustive et achevée d'un ensemble de données, mais comme les données, les documents peuvent être mémorisés, maintenus, communiqués. On désigne par le sigle GED (Gestion Electronique de Documents) l'ensemble de ces opérations. Il convient d'insister sur le caractère nouveau de ces techniques qui bouleversent actuellement le métier de documentaliste ; alors qu'autrefois, les systèmes d'information portaient seulement sur la mémorisation, la maintenance et la communication de données, les systèmes d'information multimédias portent aujourd'hui aussi sur ces mêmes opérations appliquées aux documents.



## 2 - Principes de conception



La conception d'un système d'information multimédia est soumise à des contraintes ou obligations assez fortes dont nous allons donner une liste non exhaustive. Ces contraintes proviennent d'une part, de la nature des connaissances à " enregistrer ", d'autre part aux technologies employées.

- Etant donné que les systèmes d'information multimédias sont basés sur l'utilisation d'infrastructures informatiques, on pourrait penser que leur conception est identique à celle d'un système d'information traditionnel non multimédia. Il y a effectivement bon nombre d'analogies mais aussi de nombreuses différences. On citera tout d'abord le caractère récent des applications multimédias qui ne permet pas encore d'asseoir les développements sur une expérience suffisamment longue. Les technologies utilisées sont également récentes et en constante amélioration sans encore de période de stabilité ce qui implique durant la phase de conception une veille technologique importante. Alors que le développement d'une application informatique traditionnelle ne pose généralement pas de problème en terme de maîtrise des coûts et des délais à partir d'un cahier des charges détaillé, le développement d'une application multimédia n'offre pas les mêmes garanties car l'écriture du cahier des charges suppose que le commanditaire ait une vue très précise sur ce qu'il souhaite ce qui n'est pas possible d'après les raisons exposées plus haut, mais aussi à cause de l'inexpérience de l'utilisateur final en matière de système d'information multimédia ; il y a, à notre période, beaucoup de chance (ou de malchance) pour que le système à construire et à utiliser soit le premier à être de nature multimédia. Un autre facteur de différence entre l'informatique traditionnelle et le multimédia porte sur la gestion des ressources humaines : une application multimédia fait appel à des interlocuteurs ou partenaires multiples : analystes, programmeurs multimédias, ergonomes, spécialistes du son et de l'image, designers, éditeurs, diffuseurs, financiers, fournisseurs de systèmes et d'outils, prestataires de services divers,...et commanditaires utilisateurs ce qui accroît les difficultés de communication entre ces divers professionnels et allonge les délais de réalisation. Enfin, dernier point non négligeable à signaler, pour les fournisseurs ou concepteurs, le marché des systèmes d'information multimédias est encore mal identifié et difficilement quantifiable (sauf peut-être celui des systèmes documentaires) et , pour l'essor économique des développeurs et des diffuseurs, les niches rentables ne sont pas faciles à repérer en raison des investissements élevés nécessités par les contraintes rappelées ci-dessus. Fort heureusement, il s'agit certainement d'une période transitoire qui correspond à la jeunesse, voire l'adolescence de ce secteur d'activité.
- Une phase cruciale de la conception d'un système d'information multimédia est la rédaction du cahier des charges . Il doit être le plus précis possible ce qui n'est évidemment pas simple

car l'interlocuteur commanditaire du système ne possède généralement pas l'expertise nécessaire et il faut généralement procéder par étude préalable analysant les besoins de l'utilisateur et proposant l'amorce de solutions alternatives tout en faisant œuvre de formation/information de cet utilisateur ; chaque solution doit comporter les aspects techniques, utilisation, maintenance, financiers permettant une comparaison sur ces divers critères. Un aspect important des solutions proposées doit porter sur les contenus du système d'information : qui les produit, qui les numérise, qui les enregistre, quelles charges de travail en découlent, quel volume représentent ces contenus, ... ?

- Pour faire suite au paragraphe précédent, il est utile de prévoir l'évolutivité du système d'information du point de vue des contenus. En effet, l'évaluation de ces contenus dans le temps conditionne les choix technologiques et financiers. A titre d'illustration, imaginons la construction d'un système d'information multimédia à partir de documents physiques : textes sur papier, images sous forme de photos ou de dessins sur papier, cassettes sonores, cassettes vidéos,.... Le système à construire implique la numérisation du fonds existant conduisant à un volume numérisé que l'on appellera composante de constitution ; par la suite de nouveaux documents apparaîtront qu'il faudra numériser au fur et à mesure et incorporer au système d'information, ce que l'on appellera composante d'enrichissement. Il est clair que l'évolution des volumes de la composante de constitution et de la composante d'enrichissement est différente pour chacun des deux cas et il peut être utile, en se basant sur l'activité du commanditaire du système , de modéliser cette évolution afin de prévoir à moyen terme les volumes d'information à gérer.

- Composante de constitution : elle peut correspondre à un investissement sur une période fixé pour la constitution rapide des contenus numérisés. On peut, par exemple, se baser sur un modèle exponentiel du type

$$v = v_1(1 - e^{-kt})$$

où  $v_1$  représente le volume de constitution des contenus numérisés ; le paramètre  $k$  mesure la vitesse d'acquisition ; par exemple, si l'unité de temps est l'année et si on souhaite avoir numérisé 90% du volume  $v_1$  au bout de trois ans,  $k$  vaudra 0,7675.

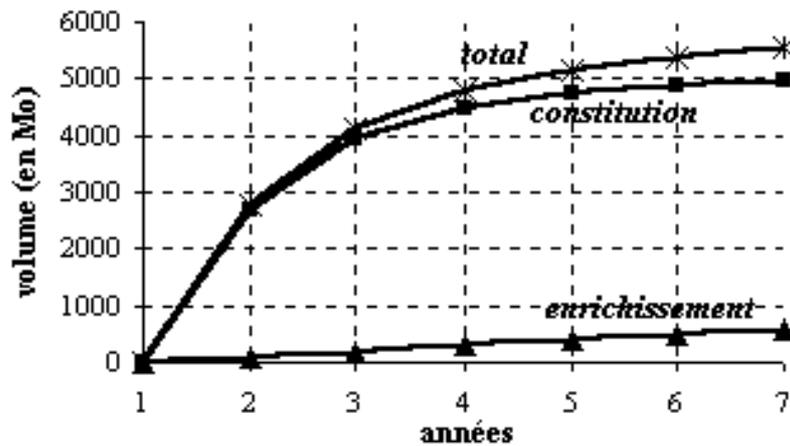
- Composante d'enrichissement : elle correspond à un fonctionnement régulier de l'acquisition des contenus numérisés et donc à un accroissement linéaire :

$$v = v_0 t$$

où  $v_0$  est la quantité de contenus numérisés acquise chaque année, par exemple 100 Mo par

an.

La superposition des deux composantes permet d'avoir une idée de l'évolution (à quelques années) du volume des contenus du système d'information.



- Les aspects ergonomiques et esthétiques sont essentiels dans une application multimédia, donc dans un système d'information multimédia. L'utilisateur moderne n'attend plus la lecture de modes d'emploi longs et compliqués ; l'utilisation d'un SIM doit être autonome, c'est à dire que l'utilisateur doit sans problème comprendre les actions possibles et être guidé par une certaine assistance incorporé au SIM. Par ailleurs, l'utilisation doit être agréable et les interfaces doivent être très soignées. Un produit multimédia est considéré comme un produit de qualité qui ne doit pas décevoir ; en effet, visualiser les tableaux d'une galerie du Louvre sur un écran implique un décor ambiant équivalent à celui que l'on peut trouver dans la galerie réelle.

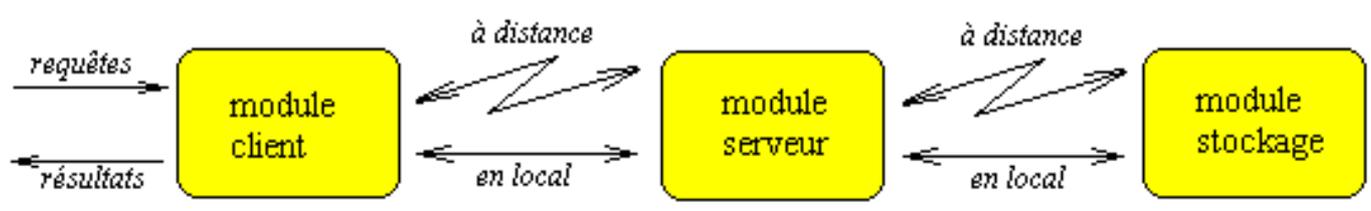


## 3 - Architecture d'un système d'information multimédia



En termes fonctionnels, un système d'information multimédia possède 3 modules essentiels :

- Le module client ou interface utilisateur qui permet d'effectuer des requêtes et de recevoir les réponses à ces requêtes ;
- Le module serveur qui reçoit les requêtes et qui met en œuvre les mécanismes de recherche d'informations pour envoi de résultats au client ;
- Le module stockage dont l'objet est la mémorisation et la maintenance des contenus numérisés : données et documents.



Il existe, bien sûr, des variantes au modèle standard présenté ci-dessus, mais la plupart des systèmes d'information multimédias est construite sur ce schéma.

Entrons dans une description plus avancée des différents modules.

### Le module client

Il peut se concevoir de diverses façons suivant que la consultation est locale ou à distance. Chaque utilisateur utilise un poste client qui n'est autre qu'un ordinateur relié au serveur par un réseau local ou longue distance. S'il s'agit d'un réseau local, une interface utilisateur est implémentée sur le poste client lui permettant de dialoguer de façon transparente à l'utilisateur avec le serveur ; cette interface peut être développée avec des outils quelconques, disponibles sur le marché des technologies. S'il s'agit d'un réseau longue distance et, a fortiori, s'il s'agit d'un système d'information ouvert à des publics variés, il vaut mieux passer par un standard que, de nos jours, les logiciels navigateurs constituent ; dans ce cas l'interface utilisateur n'est autre que le document HTML que l'on proposera au client, le navigateur se chargeant de la communication avec le serveur. Cette façon de faire se généralise d'ailleurs aussi de plus en plus pour les réseaux locaux (Intranet).

### Le module serveur

Il doit posséder des fonctionnalités de réception des requêtes, d'interprétation de celles-ci, de communication avec le module de stockage et de renvoi des résultats sous une forme acceptable par le poste client. Il doit, en outre, gérer les requêtes simultanées dans le cas où plusieurs utilisateurs, au même moment, effectuent des recherches.

Là encore, le logiciel serveur peut être développé de manière spécifique. Il peut aussi être tout simplement un serveur httpd ce qui tend à se généraliser sur Internet comme dans les Intranets. Ce serveur communique avec le module de stockage suivant un dispositif donné qui peut être, en se plaçant dans le cas d'un logiciel httpd, un programme CGI, le standard ODBC, les interfaces API. Les requêtes sont transmises au module stockage, les réponses ou résultats, plus complexes, doivent être mis en forme pour transmission au poste client.

## Le module stockage

Il constitue le noyau du système d'information car c'est lui qui gère les données et les documents et leur accessibilité. Il est usuellement constitué d'un système de gestion de base de données (SGBD) qui peut être un de ceux qui sont proposés sur le marché (relationnel ou objet) et des données et documents eux-mêmes enregistrés sur des supports adéquats. Il convient de noter que le module stockage n'est pas nécessairement en un lieu fixé, mais éventuellement partagé entre plusieurs sites géographiques (base de données répartie). Il faut noter aussi que la plupart des SGBD ne gère que les données et non des fichiers images, sons, vidéos ; il faut donc adjoindre au SGBD des modules programmés de gestion de ces nouveaux objets.



## 4 - Etudes de cas



Pour être concret, nous envisageons les trois cas suivants assez représentatifs des applications usuelles. Bornons nous ici à l'étude préalable.

### Cas 1 : le catalogue de la quincaillerie

Une quincaillerie propose 8000 articles pour les entreprises et désire que ses clients puissent effectuer leur commandes après avoir consulté un catalogue à distance donnant pour chaque article les spécifications techniques, financières, de disponibilité ; ces spécifications consistent en données, en textes et en images fixes. Les articles sont actuellement répertoriés dans une base de données classique sous Access, donnant, pour chaque article, des valeurs attachées à des champs attributs : numéro de référence, libellé, dimensions, poids, prix unitaire, disponibilité, délai de réapprovisionnement, ... Le volume de cette base de données est d'environ 500 Ko. La direction de la quincaillerie prévoit, pour mieux renseigner ses clients de fournir, pour chaque article, en plus des caractéristiques précédentes, un texte descriptif d'une douzaine de lignes de 120 caractères en moyenne et une photographie ou un schéma occupant une surface de 7 cm sur 7 cm.

Commençons par ébaucher le système d'information évoqué. La consultation se faisant à distance, on proposera d'utiliser Internet et le World Wide Web. Le client devra donc être équipé d'un micro-ordinateur relié à Internet, éventuellement via un fournisseur d'accès (provider), et pourvu d'un navigateur, si possible récent (Netscape 4, Internet Explorer 4). Le module serveur sera un serveur Web et le module stockage basé sur le SGBD d'origine Access. Le module serveur et le module stockage seront situés sur une même machine connectée évidemment à Internet.

Evaluons le volume des contenus du système d'information. A la base de données actuelle, il faudra ajouter quelques champs complémentaires (adresse des fichiers texte et image), 8000 fichiers texte, 8000 fichiers image. En ce qui concerne ces deux derniers types d'objet, on pourra considérer qu'un texte de 1000 caractères occupe 2 Ko en texte pur et qu'une image de 7 cm sur 7 cm numérisée à 100 points par pouce occupe environ 6 Ko en format GIF (un passage obligé ici pour l'utilisation sur WWW).

Le volume total à numériser est donc donné par le tableau ci-dessous :

Volume de la base de données	500 Ko
------------------------------	--------

Volume des fichiers texte	16 000 Ko
Volume des fichiers image	48 000 Ko
total	64 500 Ko

Il s'agit d'un volume de contenus assez faible (comme on verra par la suite). Le " poids " de l'image est évidemment prépondérant et il conviendra d'ajouter une marge de 20% : place occupée par les données supplémentaires à rajouter à la base et la place occupée sur le disque par les tables de pistes et de secteurs. On arrive donc à une estimation de 78 Ko.

Bien que le volume soit faible, il faut prendre les photos ou dessiner les images, ensuite les numériser. Deux possibilités existent et coexistent pour la numérisation des images. S'il s'agit de photographies, on peut les obtenir directement avec un appareil photographique numérique ou bien les faire numériser par un photographe (qui restituera un CD Photo), ou encore les numériser avec un scanner. S'il s'agit de dessins, on peut les dessiner sur papier et les passer ensuite au scanner ou bien on peut les dessiner directement avec un logiciel de dessin. Tous ces travaux devront être quantifiés en temps et en coûts. Par ailleurs les textes descriptifs des articles doivent être écrits ; ils peuvent l'être directement à l'aide d'un traitement de texte, mais le problème ici est le style de cette écriture car il ne faut pas oublier qu'il s'agit également d'un outil de promotion de l'entreprise ce qui implique un soin certain dans l'écriture..

Quant à l'évolution du système, il est fortement conditionné par l'acquisition des images et la confection des textes. Si on se donne une durée d'un an pour aboutir à la composante de constitution, il faut bien prendre en considération la nécessité de prendre/dessiner 8000 photos/schémas et d'écrire 8000 textes ; dans cette tâche le commanditaire doit nécessairement s'impliquer. La composante d'enrichissement se limite à une maintenance du système en terme de contenus ; on peut par exemple considérer que 10% des articles sont modifiés par an (remplacement d'articles par d'autres) ce qui nécessite, certes, une mise à jour de la base de données, mais surtout le remplacement de 800 textes et de 800 images. Le commanditaire doit être conscient de ces charges de maintenance car un système qui n'est pas mis à jour est très vite déconsidéré.

## Cas 2 : le syndicat d'initiative

Le syndicat d'initiative d'une ville s'intéresse à procurer aux touristes éventuels un dispositif d'information sur les monuments à visiter (édifices, statues, parcs, musées,...) et envisage un système d'information consultable à distance et donnant sur requête des informations sur les sites intéressants sous forme de fiches proposant un descriptif du site (texte), une ou plusieurs photos (images fixes), éventuellement de courtes vidéos et de courts commentaires sonores.

Ce syndicat d'initiative (le mot initiative prend, dans ce cas, toute sa valeur sémantique) propose de répertorier 50 sites à terme. Il conviendra donc d'écrire 50 notices explicatives textuelles, de prendre au moins 50 photos, et au plus 200 photos (4 par site), d'acquérir au plus 50 vidéos de 30 secondes et 50 commentaires sonores de 30 secondes également. On peut considérer que chaque notice comprend l'équivalent d'une demi-page A4 remplie, soit environ 3 Ko en texte pur, que 1 image correspond à 10 Ko en format GIF, que 1 minute de vidéo (sans son, 1/9 d'écran) correspond en format AVI à 50 Mo et que 1 seconde de son correspond en format AU à 175 Ko. Examinons avec ces hypothèses le volume à enregistrer en discernant la configuration initiale (50 notices textuelles, 50 photos) et la configuration définitive (50 notices textuelles, 200 photos, 50 vidéos, 50 commentaires sonores) :

Médias (configuration initiale)	taille
50 notices textuelles	150 Ko
50 images fixes	500 Ko
total	650 Ko
Médias (configuration définitive)	taille
50 notices textuelles	150 Ko
200 images fixes	2 000 Ko
50 vidéos	1 250 000 Ko
50 commentaires sonores	262 500 Ko
total	1 514 650 Ko

Pour cet exemple, on s'aperçoit que le volume des contenus devient important. On remarquera tout particulièrement la capacité d'enregistrement qui doit être dédiée à la vidéo et au son. Il convient, dans ce cas particulier, d'avertir ce syndicat d'initiative sur les difficultés de communication (actuelles) de la vidéo et de l'image à distance. En effet, comme il s'agit de fournir ces informations aux touristes intéressés et que ceux-ci, dans le pire des cas, recevront ces informations chez eux sur un poste client équipé d'un modem à 28 800 bits/s, il faut évaluer le temps de téléchargement d'une seule vidéo à près de 2 h ! Si par contre, l'utilisateur dispose d'une liaison à 2 Mbits/s, cas exceptionnel pour un particulier, le temps de téléchargement se réduit à 1,6 minute. Devant cette situation, il est clair que, à moins de réduire drastiquement la longueur temporelle des vidéos, ce qui en enlève l'intérêt, il conviendra de se passer des vidéos à distance et attendre l'avènement de réseaux publics de plus hauts débits.

L'infrastructure à mettre en œuvre est assez semblable à celle du cas n°1 : le poste client est un poste banalisé sur réseau Internet avec un logiciel navigateur. ; le serveur est un serveur Web et la base de données peut être construite sur un SGBD du commerce, les images et les notices étant des fichiers repérés dans la base de données par leurs adresses.

### Cas n° 3 : Médiathèque sur le patrimoine

Une collectivité locale souhaite enregistrer des éléments d'histoire de son territoire et a procédé à des interviews chez les particuliers, particulièrement les particuliers âgés, pour recueillir des informations sur la vie d'autrefois, les petits métiers des campagnes, les fêtes de village, etc... A l'occasion des interviews, des photos sont prises et des documents divers sont récoltés. Tous ces matériaux (images, photos, documents papiers, cassettes audio) sont répertoriés par lieu, date, thèmes et archivés manuellement par des méthodes traditionnelles. Devant l'accumulation de ces objets, il est décidé d'informatiser le fonds pour des raisons de facilité de classement, mais aussi et surtout, pour permettre une consultation aisée suivant plusieurs critères.

Le fonds correspond aux quantités suivantes :

Images : environ 600 photographies

Enregistrements sonores : environ 100 cassettes d'une heure approximativement.

Textes : fiches d'accompagnement des enregistrements sonores (1 fiche par enregistrement occupant 1 à 3 pages A4).

Par ailleurs, le fonds s'accroît chaque année de nouvelles acquisitions : environ 50 photographies et 10 cassettes audio.

En prenant pour base les correspondances suivantes, 1 page A4 = 4 Ko, 1 h de son = 630 Mo, 1 photographie = 100 Ko (moyenne très contestable, la taille dépendant du nombre de pixels, du nombre de couleurs et du format utilisé), on obtient les résultats très approximatifs suivants :

Médias	taille
images	60 000 Ko
enregistrements sonores	63 000 000 Ko
textes	800 Ko
total	63 060 800 Ko

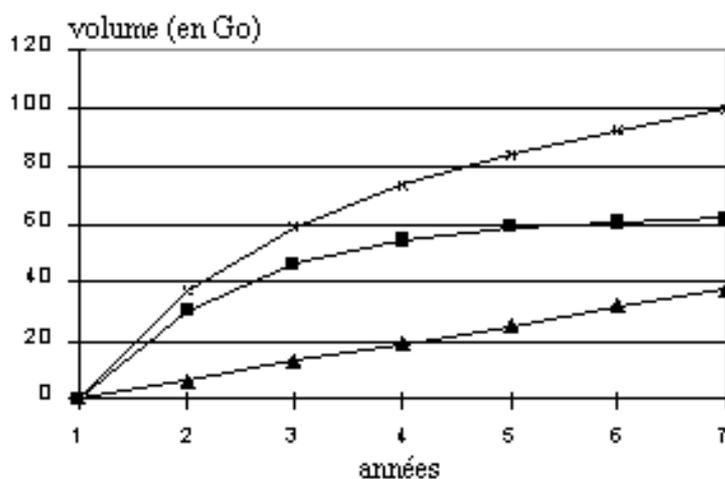
Le volume est ici très impressionnant dû essentiellement à la numérisation des enregistrements

sonores : 63 Go. Ce volume est évidemment susceptible d'augmentation, car de nouveaux enregistrements sonores s'ajouteront au cours des années et devront à leur tour être numérisés. On peut effectuer une simulation sur l'évolution des volumes au cours du temps. Pour rester vraisemblable, on ne dépassera pas 6 années dans cette simulation et on admettra que l'on se donne 3 années pour numériser 90% du fonds existant tout en procédant parallèlement à la numérisation régulière des nouvelles acquisitions, ce qui représente approximativement 6,3 Go par an. Appliquons le modèle décrit plus haut pour les composantes de constitution et d'enrichissement.

Composante de constitution :  $v=63(1-e^{-0,77t})$

Composante d'enrichissement :  $v=6,3t$

La figure ci-dessous indique l'évolution du fonds au cours des six premières années.



L'infrastructure informatique à mettre en place peut être esquissée sous la forme d'un réseau local de postes de consultation, par exemple bus fast Ethernet 100 Mbits/s. Le serveur sera doté d'un dispositif de stockage assez conséquent, constitué de disques RAID, qui semblent, pour le moment, constituer la meilleure solution (l'utilisation d'un juke box de CD-ROM peut constituer une alternative, toutefois moins performante en raison des temps d'accès). Par ailleurs, les cassettes analogiques ayant la fâcheuse tendance à s'effacer progressivement, il sera utile de procéder à des copies de conservation sur CD-ROM (dont la durée de conservation annoncée est de 100 ans). Un équipement de numérisation sera également à prévoir si la médiathèque effectue elle-même l'acquisition numérique du son et de l'image ; un prestataire extérieur peut aussi être prévu pour cette tâche ; une étude des coûts devrait trancher à ce sujet.

Au niveau de la gestion des médias, un SGBD du commerce pourrait être utilisé, relationnel ou orienté objet.



## 5 - Bibliographie



- |                        |  |          |
|------------------------|--|----------|
| S. MIRANDA, J-M. BUSTA | L'art des bases de données,<br>tome 1 : Introduction aux bases<br>de données | Eyrolles |
| R.M. KAPLAN            | Intelligent Multimedia Systems   | Wiley    |
| R. BERNARD             | L'Intranet en entreprise   | Sybex    |



# Gestion de l'information multimédia

## Conception et Réalisation

---

Objectifs

Introduction

1. Conception : cas n°1
2. Conception : cas n°2
3. Conception : cas n°3
4. Réalisation : cas n°1
5. Réalisation : cas n°2
6. Réalisation : cas n°3

 Téléchargement du chapitre (chap7.pdf - ... Ko)

---

Concepteurs : 

Gérard-Michel Cochard

# Introduction

---



Ce chapitre reprend les trois exemples du chapitre précédent et, pour chacun d'eux, illustre l'application des techniques de conception et des éléments de réalisation. Les exemples présentés sont tirés de situations réelles ; nous avons simplifié ces situations pour que l'exposé reste didactique.

---

⇒ Sections du chapitre : [1](#) | [2](#) | [3](#) | [4](#) | [5](#) | [6](#) | [7](#)

# 1 - Conception - Cas n° 1 :

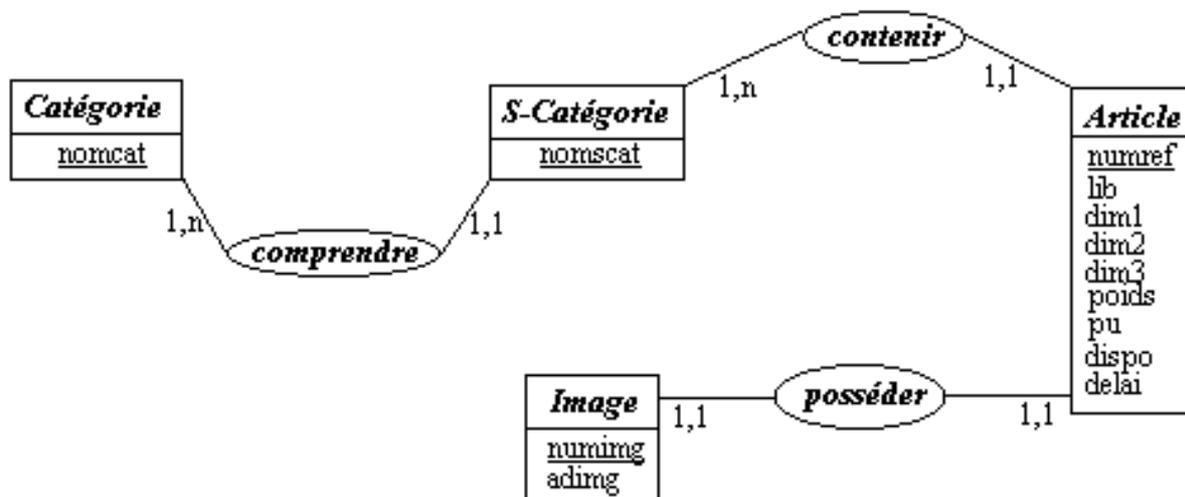
## le catalogue de la quincaillerie



### Modèle conceptuel des données

Le modèle est assez simple étant donné le système de classement de la quincaillerie : les articles sont classés en catégories et sous-catégories. Un article appartient à une sous-catégorie et une seule et une sous-catégorie appartient à une catégorie et une seule. Un article est identifié par un numéro de référence unique et ses propriétés sont un libellé descriptif, des dimensions spatiales, un poids, un prix unitaire, une disponibilité (oui/non), un délai de réapprovisionnement et l'adresse du fichier image correspondant.

Par suite, il est aisé d'imaginer les entités à prendre en compte dans le formalisme entité-association : Catégorie, Sous-catégorie, Article, Image. Les associations et cardinalités sont évidentes compte tenu des explications précédentes. Le schéma conceptuel est donné à la figure ci-dessous :



### Modèle logique des données

Le passage au modèle relationnel est effectué par application des règles de passage mentionnées au chapitre 3 :

Règle R1 : les entités deviennent des tables ; les identifiants deviennent des clés ; on obtient ainsi les tables :

CATEGORIE (nomcat)

SCATEGORIE (nomscat)

ARTICLE (numref, lib, dim1, dim2, dim3, poids, pu, dispo, delai)

IMAGE (numimg, adimg)

Règle R2 : on recherche les cardinalités 1,1 et on ajoute un attribut supplémentaire dans la table correspondante : la clé de la table associée :

SCATEGORIE(nomscat, #nomcat)

ARTICLE (numref, lib, dim1, dim2, dim3, poids, pu, dispo, delai, #numimg, #nomscat)

IMAGE (numimg, adimg, #numref)

Règle R3 : recherche des associations possédant des cardinalités  $x, n-y, n$ . Il n'y en a pas ici.

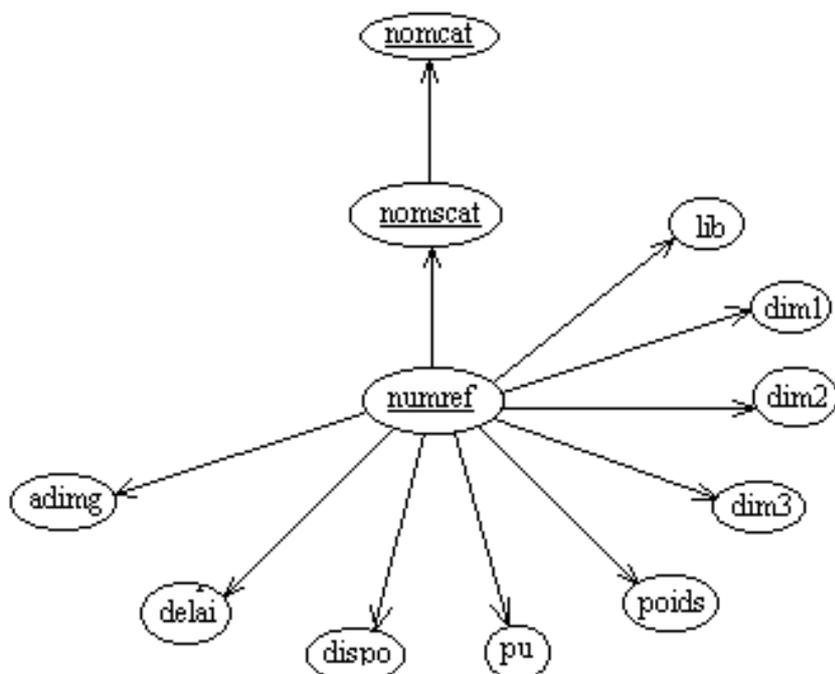
Finalement, on peut dresser la liste des tables du modèle relationnel. Toutefois, l'analyse des entités ou des tables obtenues montre que l'on peut supprimer la table (et l'entité image) à cause de l'association "posséder" qui possède les cardinalités 1, 1-1, 1, de sorte que l'on peut supprimer l'attribut numimg et remplacer celui-ci, dans la table ARTICLE par adimg :

CATEGORIE (nomcat)

SCATEGORIE(nomscat, #nomcat)

ARTICLE (numref, lib, dim1, dim2, dim3, poids, pu, dispo, delai, adimg, #nomscat)

Le graphe des dépendances fonctionnelles exprime graphiquement le modèle précédent :



## Modèle théorique des traitements

Le traitement unique est la recherche d'un article, ce qui peut se faire, soit par sa référence, si elle est connue de l'utilisateur (supposition justifiée par le fait que la quincaillerie possède une clientèle régulière d'entreprises), soit par catégorie et sous-catégorie.

T1 : recherche par numéro de référence

```

Boucle :
    Saisie du numéro de référence
    Recherche et affichage du document composite
    correspondant à l'article
Fin Boucle
  
```

T2 : recherche par catégorie et sous catégorie

```

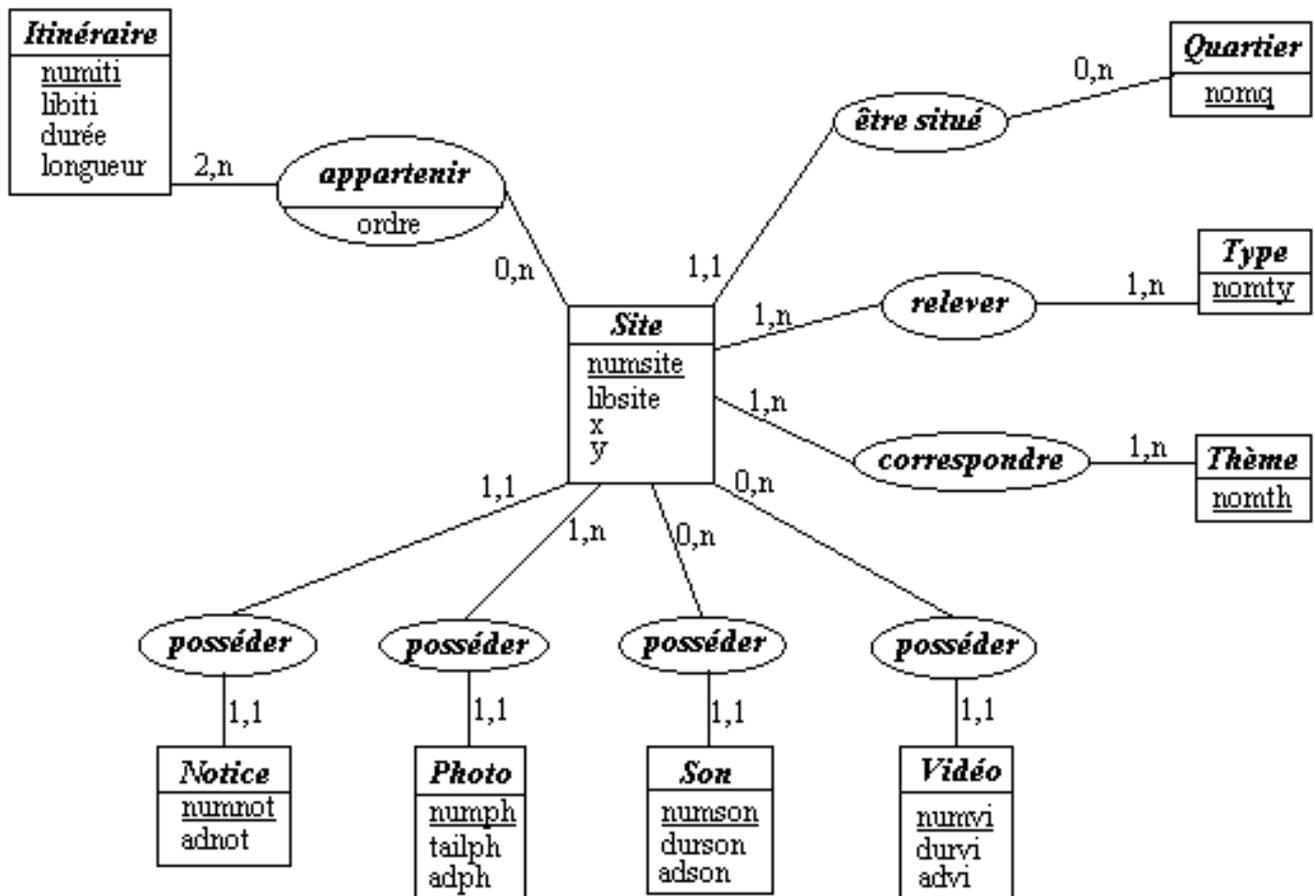
Affichage de la liste des catégories
Boucle :
    Choix d'une catégorie
    Affichage de la liste des sous-catégories correspondantes
    Boucle :
        Choix d'une sous-catégorie
        Affichage de la liste des articles correspondants
        Boucle :
            Choix d'un article
            Affichage du document composite
correspondant au choix.
        Fin Boucle
    Fin Boucle
Fin Boucle
  
```



## 2 - Conception - Cas n° 2 : le syndicat d'initiative

### Modèle conceptuel des données

L'analyse conduite a discerné comme centrale l'entité Site, et des entités secondaires : l'itinéraire, Thème, Quartier, Type, Notice, Image, Vidéo, Son. Ces deux dernières entités sont prises en compte dans le but de préserver l'avenir bien que dans l'implémentation à court terme, il n'y aura ni vidéo, ni commentaire sonore.



La figure ci-dessus présente le modèle conceptuel des données dans le formalisme entité-association. Un site est repéré par un numéro d'identification, un libellé descriptif sommaire et des coordonnées sur une carte. Un itinéraire est identifié par un numéro et caractérisé par un libellé descriptif, une durée et une longueur de parcours. Un site qui se trouve sur un itinéraire est affecté d'un numéro d'ordre de parcours sur cet itinéraire. Un quartier est identifié par son nom, un type par un nom de type, un thème par un nom de thème. Une notice, une photo, un son, une vidéo sont identifiés par un numéro unique ; une photo peut être caractérisée par sa taille, un son et une

vidéo par leur durée. Les propriétés adnot, adph, adson, advi désignent les adresses des fichiers texte (pour les notices), des fichiers image (pour les photos ou les dessins), des fichiers son (pour les plages sonores), des fichiers vidéo (pour les vidéos).

Ce modèle appelle quelques commentaires, en particulier sur les cardinalités. Ces dernières s'expliquent par les règles suivantes (définies d'après l'activité du commanditaire) : tout itinéraire touristique comporte au moins 2 sites, mais un site peut ne pas se trouver sur un itinéraire ; un site est situé dans un quartier et un seul, mais un quartier peut ne pas comporter de site ; un site relève au moins d'un type (cathédrale, musée, château, ...) ; un site correspond au moins à un thème (gothique, guerre de cent ans, révolution, baroque, ...) ; un site possède une et une seule notice, une ou plusieurs photos, éventuellement une ou plusieurs vidéos, éventuellement un ou plusieurs plages sonores, mais une notice correspond à un site et un seul et de même pour une photo, une vidéo, un son.

## Modèle logique des données

Appliquons les règles de passage du modèle entité-association au modèle relationnel :

Règle 1 : toutes les entités deviennent des tables, l'identifiant devient la clé ; on obtient ainsi les tables suivantes :

SITE(numsite, libsite, x, y)  
 ITINERAIRE (numiti, libiti, duree, longueur)  
 QUARTIER (nomq)  
 TYPE (nomty)  
 NOTICE (numnot, adnot)  
 PHOTO (numphoto, tailph, adph)  
 SON (numson, durson, adson)  
 VIDEO (numvi, durvi, advi)

Règle 2 : on recherche les cardinalités 1, 1 et on incorpore à la table un attribut supplémentaire consistant en la clé de la table reliée (clé externe) :

SITE (numsite, libsite, x, y, #numnot, #nomq)  
 NOTICE (numnot, adnot, #numsite)  
 PHOTO (numph, tailph, adph, #numsite)  
 SON (numson, durson, adson, #numsite)  
 VIDEO (numvi, durvi, advi, #numsite)

Règle 3 : on recherche les cardinalités x, n—y, n et on crée des tables nouvelles à partir des associations correspondantes :

PARCOURS (numsite, numiti, ordre)

INDEX1 (numsite, nomty)

INDEX2 (numsite, nomth)

On peut remarquer, par ailleurs, que la table NOTICE est inutile en raison des cardinalités 1, 1—1, 1 entre les entités Notice et Site ; il suffit d'ajouter l'attribut adnot dans la table SITE en remplacement de la clé externe numnot. En définitive notre modèle relationnel est constitué des tables suivantes :

SITE (numsite, libsite, x, y, adnot, #nomq)

ITINERAIRE (numiti, libiti, duree, longueur)

QUARTIER (nomq)

THEME (nomth)

TYPE (nomty)

PHOTO (numph, tailph, adph, #numsite)

SON (numson, durson, adson, #numsite)

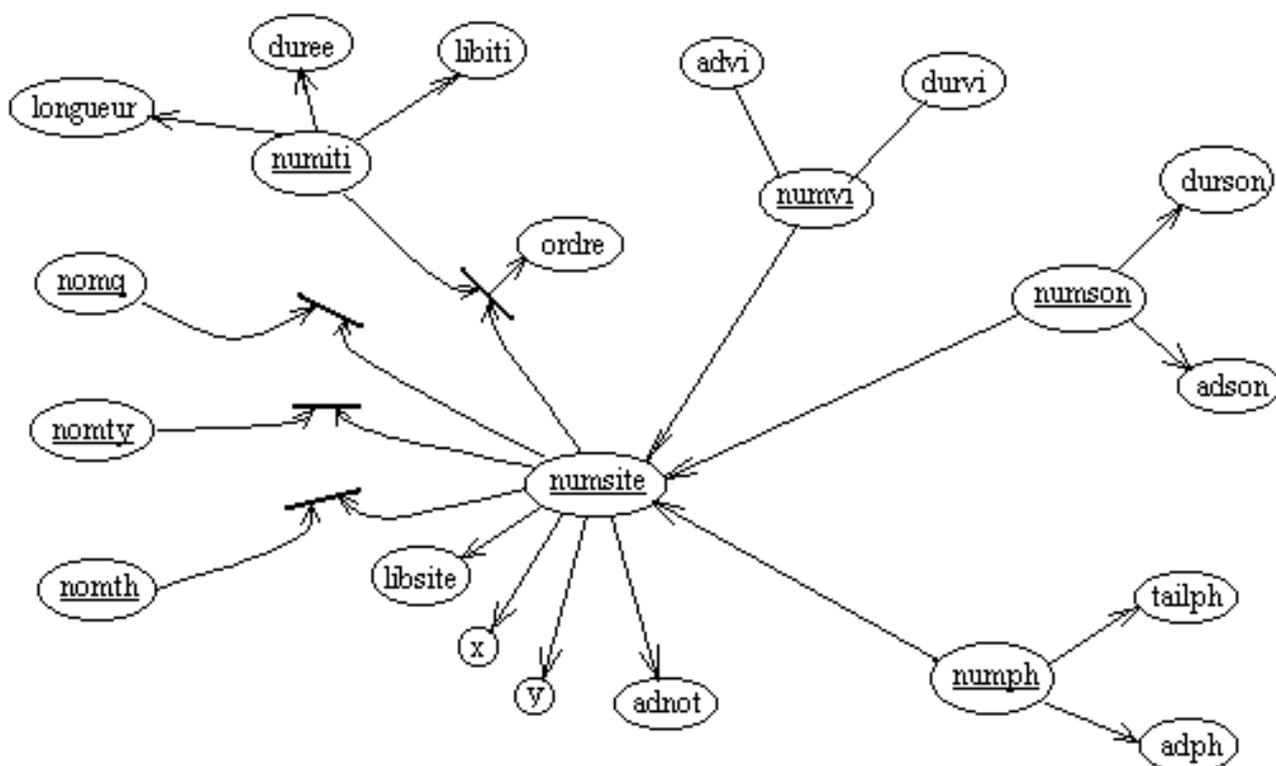
VIDEO (numvi, durvi, advi, #numsite)

PARCOURS (numsite, numiti, ordre)

INDEX1 (numsite, nomty)

INDEX2 (numsite, nomth)

Cette situation peut être représentée par le graphe des dépendances fonctionnelles.



## Modèle théorique des traitements

Deux traitements principaux sont à analyser :

- T1 : Recherche d'un site ; cette recherche peut s'opérer par thème, par quartier ou par type.

T11 : recherche d'un site par thème :

Affichage de la liste des thèmes

Boucle :

    Saisie du thème

    Fourniture de la liste des sites correspondants

    Boucle :

        Choix d'un site dans la liste

        Affichage du document composite relatif au site choisi

    Fin Boucle

Fin Boucle

T12 : recherche d'un site par quartier

Affichage de la liste des quartiers

Boucle :

    Saisie du quartier

    Fourniture de la liste des sites correspondants

    Boucle :

        Si la liste n'est pas vide, choix d'un site dans la liste

        Affichage du document composite relatif au site choisi

    Fin Boucle

Fin Boucle

T13 : recherche d'un site par type

Affichage de la liste des types

Boucle :

    Saisie du type

    Fourniture de la liste des sites correspondants

    Boucle :

        Choix d'un site dans la liste

        Affichage du document composite relatif au site choisi

    Fin Boucle

Fin Boucle

- T2 : recherche d'un itinéraire

Fourniture de la liste des itinéraires et de leur descriptif

Boucle :

    Choix d'un itinéraire dans la liste

Affichage du document composite relatif à l'itinéraire choisi

Boucle :

    Si choix d'un site, affichage du document composite du site  
choisi.

    Fin Boucle

Fin Boucle



## 3 - Conception - Cas n° 3 : médiathèque sur le patrimoine



Comme les autres cas, celui de la médiathèque sur le patrimoine est tiré d'une étude réelle ; pour les besoins de l'exposé, cette étude a été simplifiée car l'objet de ce cours est avant tout didactique.

### Modèle conceptuel des données

De la manière actuelle de travailler de la médiathèque, on peut distinguer trois niveaux de classement des matériaux récoltés durant les interviews : la collecte, l'objet (pris ici dans un sens non technique), l'item. Une collecte est un ensemble d'objets ; les objets sont les enregistrements audio, les images (photos, dessins), les brochures papier, etc... ; un objet est composé d'items : une brochure papier comporte des items texte, des items image, de même un enregistrement audio peut être décomposé en items sonores, chacun d'eux ayant une consistance propre (en effet, dans une conversation, plusieurs sujets peuvent successivement être abordés, ils correspondront chacun à un item).

Les responsables de la médiathèque désirent également gérer les utilisateurs, les personnes morales ou physiques impliquées dans les collectes, objets, items.

Plus précisément une collecte sera numérotée, correspondra à un titre, une date et comportera un résumé. Un objet sera également numéroté et comportera de nombreuses propriétés : type, libellé, support, renseignements techniques (taille d'image, format, durée d'enregistrement, ...) que nous ne détaillerons pas ici par souci de simplification ; toutefois, tous les objets auront le même nombre de propriétés mais celles-ci différeront en signification suivant le type d'objet et le support. Un item portera un numéro d'identification, un libellé, un type, des propriétés techniques (en nombre égal pour tous les items).

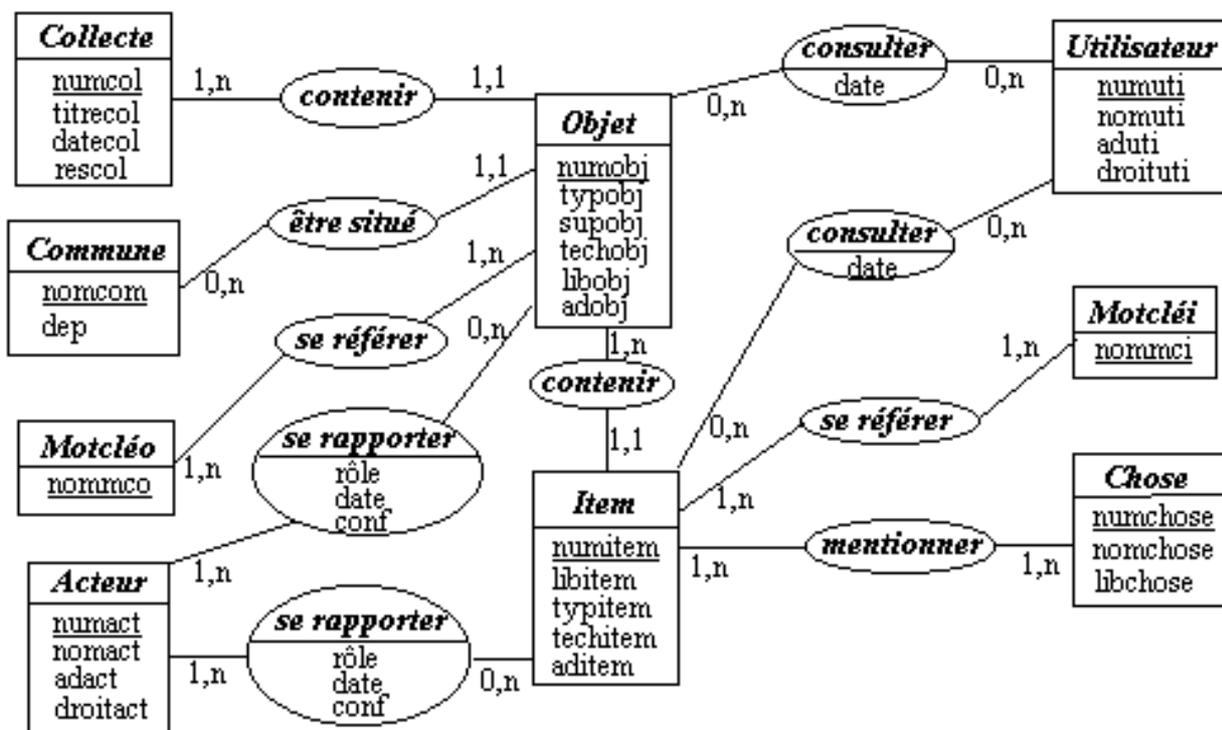
En ce qui concerne les personnes, on peut distinguer les utilisateurs qui consultent le fonds et les acteurs impliqués dans le fonds (participant, personne qui parle, personne qui figure sur une photo, propriétaire de l'objet). Chacune de ces personnes porte un numéro matricule, un nom ou une raison sociale, une adresse, des droits d'accès ou de propriété.

On peut donc distinguer les entités suivantes : Collecte, Objet, Item, Acteur, Utilisateur. Il convient d'y ajouter des informations relatives à la localisation, par exemple la commune, d'où une entité additionnelle : Commune.

Par ailleurs, pour les besoins de la recherche, il est utile de définir des mots-clés permettant de retrouver des objets ou des items particuliers. Après réflexion, deux classes de mots clés ont été définies : celle qui permet de retrouver les objets et celle, plus fine, qui permet de retrouver les items. On ajoute donc deux autres entités : Motcléo et Motcléi.

Enfin, après une réflexion plus poussée, il a paru intéressant de rajouter une dernière entité : la Chose relative aux objets réels mentionnés au niveau des items.

En définitive, le modèle entité-association possède la physionomie illustrée dans la figure ci-dessous :



Les propriétés utilisées ont la signification suivante :

titrecol : titre de la collecte, datecol : date de la collecte, rescol : résumé de la collecte, dep : département, nomact : nom de l'acteur, adact : adresse de l'acteur, droitact : droits attachés à l'acteur, rôle : rôle de l'acteur, date : date relative au rôle effectué par l'acteur, conf : niveau de confidentialité, typobj : type d'objet, supobj : support de l'objet, technobj : spécificités techniques de l'objet, libobjet : libellé de l'objet, adobj : adresse du document objet, libitem : libellé de l'item, typitem : type de l'item, techitem : spécificités techniques de l'item, aditem : adresse du fichier de l'item, nomuti : nom de l'utilisateur, aduti : adresse de l'utilisateur, droituti : droits d'accès de l'utilisateur, nomchose : nom de la chose, libchose : libellé descriptif de la chose, date : date de consultation.

Nous insistons sur le fait que certaines propriétés regroupent en fait plusieurs propriétés distinctes dans un but de lisibilité et de simplicité. Ainsi techitem et techobj correspondent à un

ensemble de propriétés techniques des items et des objets, adact et aduti désignent l'ensemble des propriétés d'adressage : voie, numéro de voie, code postal, ville, téléphone, télécopie, e-mail, etc...

## Modèle logique des données

Appliquons les règles usuelles qui permettent de passer du modèle conceptuel dans le formalisme entité-association au modèle logique dans le formalisme relationnel.

Règle 1 : les entités deviennent des tables et les identifiants deviennent des clés ; les tables issues de cette règle sont données ci-dessous :

COLLECTE (numcol, titrecol, datecol, rescot)  
 OBJET (numobj, typobj, supobj, techobj, libobj)  
 ITEM (numitem, libitem, typitem, techitem)  
 ACTEUR (numact, nomact, adact, droitact)  
 UTILISATEUR (numuti, nomuti, aduti, droituti)  
 CHOSE (numchose, nomchose, libchose)  
 COMMUNE (nomcom, dep)  
 MOTCLEO (nommco)  
 MOTCLEI (nommci)

Règle 2 : on recherche les associations avec des cardinalités 1, 1 et on introduit la clé de la table associée dans la table correspondante :

OBJET (numobj, typobj, supobj, techobj, libobj, #nomcom, #numcol)  
 ITEM (numitem, libitem, typitem, techitem, #numobj)

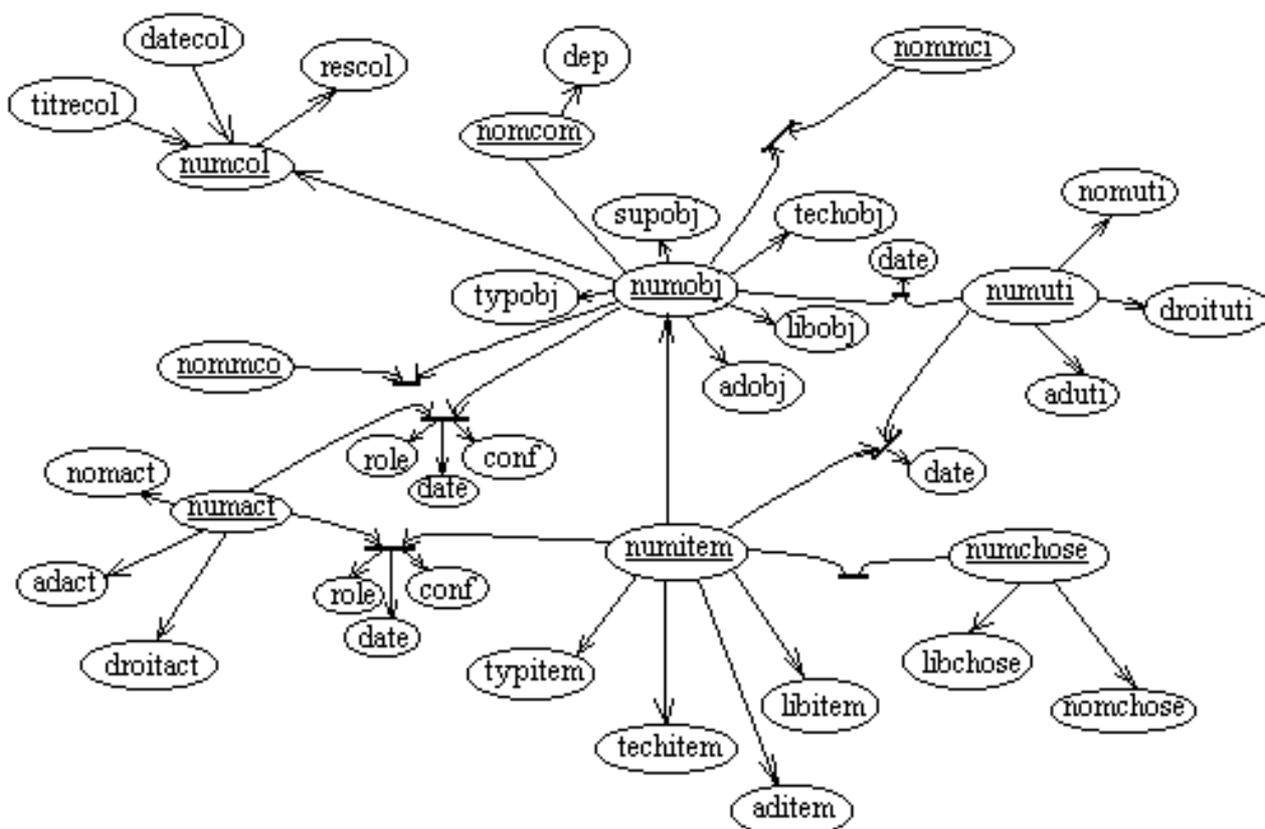
Règle 3 : les associations qui possèdent des cardinalités de type x, n—y, n deviennent des tables dont la clé est composée des clés des tables associées et dont les propriétés sont celles de l'association :

INDEXO (nommco, numobj)  
 INDEXI (nommci, numitem)  
 UTIOBJ (numuti, numobj, date)  
 UTIITE (numuti, numitem, date)  
 ACTOBJ (numact, numobj, rôle, date, conf)  
 ACTITE (numact, numitem, rôle, date, conf)  
 CHOITE (numchose, numitem)

En définitive, le modèle relationnel est composé des tables suivantes :

COLLECTE (numcol, titrecol, datecol, rescol)  
 OBJET (numobj, typobj, supobj, techobj, libobj, #nomcom, #numcol)  
 ITEM (numitem, libitem, typitem, techitem, #numobj)  
 MOTCLEO (nommco)  
 MOTCLEI (nommci)  
 ACTEUR (numact, nomact, adact, droitact)  
 UTILISATEUR (numuti, nomuti, aduti, droituti)  
 CHOSE (numchose, nomchose, libchose)  
 COMMUNE (nomcom, dep)  
 INDEXO (nommco, numobj)  
 INDEXI (nommci, numitem)  
 UTIOBJ (numuti, numobj, date)  
 UTIITE (numuti, numitem, date)  
 ACTOBJ (numact, numobj, rôle, date, conf)  
 ACTITE (numact, numitem, rôle, date, conf)  
 CHOITE (numchose, numitem)

Il correspond au graphe des dépendances fonctionnelles décrit par la figure suivante :



## Modèle théorique des traitements

Comme pour les études de cas précédentes, nous nous intéresserons ici uniquement aux traitements relatifs à la consultation. De nombreux traitements peuvent être prévus sur la base de données ci-dessus. Nous en donnons seulement quelques exemples (susceptibles chacun de variantes diverses) :

## T1 : consultation d'une collecte

Affichage de la liste des collectes

Boucle :

Choix d'une collecte

Recherche et affichage de la liste des objets relatif à la collecte choisie

Boucle :

Choix d'un objet dans cette liste

Recherche et Affichage du document composite relatif à cet objet avec

bouton "son" éventuellement

Boucle :

Diffusion du son si le bouton "son" est actionné

Fin Boucle

Fin Boucle

Fin Boucle

## T2 : recherche par mot clé d'objets

Affichage de la liste des mots clés

Boucle :

Choix d'un mot clé

Recherche et Affichage de la liste des objets relatifs à ce mot clé

Boucle :

Choix d'un objet dans cette liste

Recherche et Affichage du document composite relatif à cet objet avec

bouton "son" éventuellement

Boucle :

Diffusion du son si le bouton "son" est actionné

Fin Boucle

Fin Boucle

Fin Boucle

## T3 : recherche par mot clé d'items

Affichage de la liste des mots clés

Boucle :

Choix d'un mot clé

Recherche et Affichage de la liste des items relatifs à ce mot clé

Boucle :

Choix d'un item dans cette liste

Recherche et Visualisation/audition de l'item

Fin Boucle

Fin Boucle

T4 : recherche par localisation

Affichage de la liste des communes

Boucle :

Choix d'une commune

Recherche et Affichage de la liste des objets relatifs à cette commune

Boucle :

Choix d'un objet dans cette liste

Recherche et visualisation/audition de cet objet

Fin Boucle

Fin Boucle

T5 : recherche d'un utilisateur

Affichage de l'alphabet

Boucle :

Choix de la lettre initiale du nom de l'utilisateur

Recherche et Affichage de la liste des utilisateurs répondant à ce critère

Boucle :

Choix d'un utilisateur

Recherche et Affichage des caractéristiques de cet utilisateur avec bouton

"utilisation"

Si le bouton "utilisation" est activé, recherche et affichage de l'historique des consultations.

Fin Boucle

Fin Boucle



## 4 - Réalisation - cas n° 1 :

### le catalogue de la quincaillerie



Le module serveur et le module stockage seront réalisés à partir d'un micro-ordinateur en connexion sur Internet et muni

- d'un logiciel serveur WWW httpd
- d'un SGBD relationnel du commerce (Oracle, Access, SQL Server, ...)

Sur le disque dur C: de cette machine, seront enregistrés

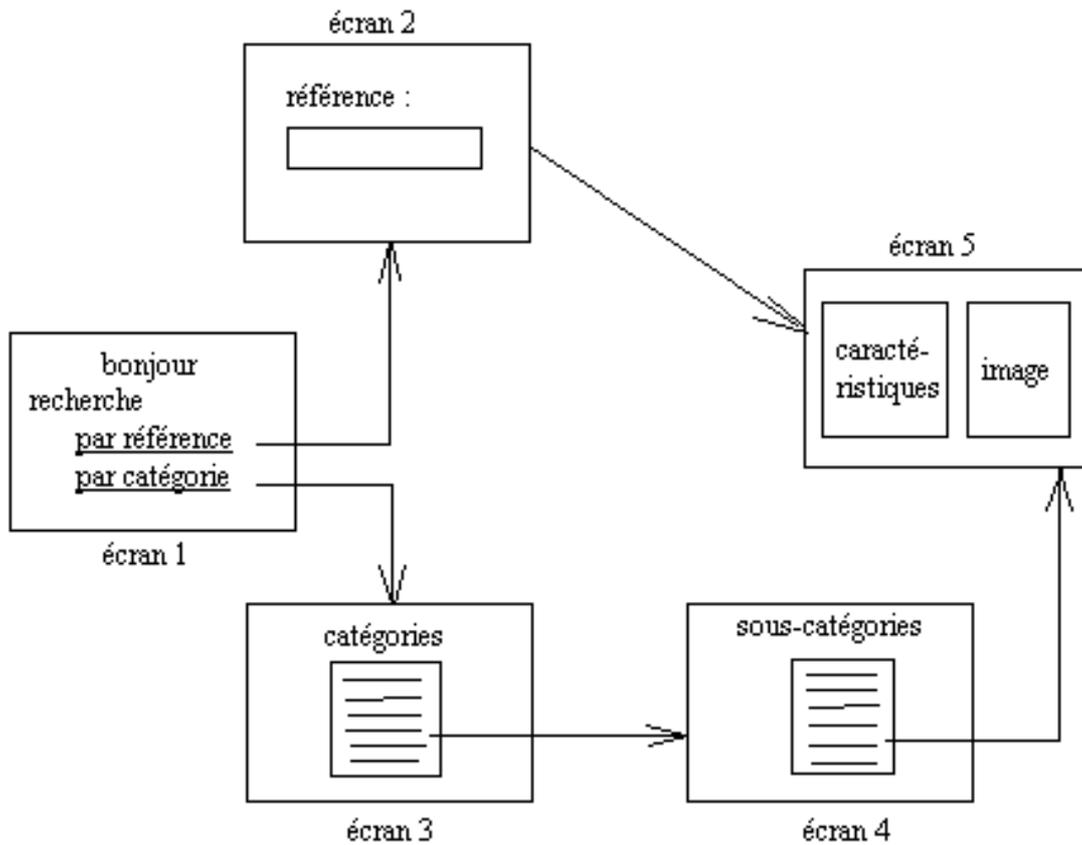
- les fichiers images dans le répertoire C:\images
- la base de données dans le répertoire C:\base
- le site serveur dans le répertoire C:\Web\root\quincaill

Bien entendu, ces répertoires sont totalement fictifs ici. Leurs noms sont donnés pour la compréhension de la suite.

La connexion avec le serveur est opérée à partir d'un poste client doté d'un logiciel navigateur. L'appel s'opère par la saisie de l'URL du site : [http://nom\\_serveur/quincaill](http://nom_serveur/quincaill). Cet appel génère l'envoi d'un document HTML d'accueil proposant la recherche d'un article, soit par référence, soit par catégorie et sous-catégorie. Le choix de l'utilisateur détermine par hyperlien l'envoi des documents HTML adéquats :

- page de saisie d'un numéro de référence : le choix renvoie au serveur le critère qui servira à la recherche dans la base de données et la récupération des caractéristiques correspondantes de l'article, y compris l'adresse du fichier image. Le tout sera renvoyé sous forme d'un document HTML.
- ou bien page de choix d'une catégorie ; le choix d'une catégorie est un critère permettant de retrouver dans la base toutes les sous-catégories correspondantes et d'en envoyer l'affichage dans le cadre d'un document HTML actif. Le choix d'une sous-catégorie envoie à son tour le critère de recherche des articles qui lui correspondent ; leur liste est renvoyée dans un document HTML actif. Le choix d'un article dans cette liste provoque la recherche dans la base de données des caractéristiques de cet article et le document HTML composite portant sur cet article (caractéristiques + image) est renvoyé au client.

On peut imaginer, pour l'utilisateur la suite d'écrans suivante :



Les problèmes qui se posent dans ce mode opératoire sont les suivants :

P1 : lien entre l'action de l'utilisateur et la recherche dans la base de données

P2 : restitution du résultat

Prenons ces problèmes dans cet ordre.

**P1** : Connexion aux sources de données.

L'utilisateur agit/choisit par clic sur ce qu'il désire. Ceci se traduit,

- soit par l'utilisation d'une balise <A> avec l'attribut href = "url" où "url" désigne l'adresse d'un document HTML ou une requête (programme CGI ou fichier IDC)
- soit par l'utilisation d'un formulaire et de l'attribut action = "url" où "url" est l'adresse d'une requête (programme CGI ou fichier IDC)

Dans tous les cas, il faut prévoir une requête en SQL chaque fois que l'on a besoin d'effectuer des recherches dans la base de données. Par exemple :

T1 : recherche à partir d'un numéro de référence num :

```
SQL1 :  SELECT numref, lib, dim1, dim2, dim3, poids, dispo, delai, adimg
        FROM ARTICLE
        WHERE numref = num ;
```

T2 : recherche par catégorie et sous-catégorie

-détermination de la liste des catégories

```
SQL2 :  SELECT *
        FROM CATEGORIE ;
```

-détermination de la liste des sous-catégories à partir du choix d'une catégorie cat :

```
SQL3 :  SELECT nomscat
        FROM SCATEGORIE
        WHERE nomcat = cat ;
```

-détermination des articles appartenant à une sous-catégorie définie par scat :

```
SQL4 :  SELECT numref, lib
        FROM ARTICLE
        WHERE nomscat = scat ;
```

-détermination d'un article particulier de numéro de référence donné : se rapporter au traitement T1.

Comment peut-on communiquer ces requêtes à la base de données à partir du poste client ? Un moyen simple est l'utilisation d'ODBC. Pour cela il faut procéder comme suit :

1) déclarer, dans le pilote ODBC, la base de données supposée être réalisée avec Access. Ouvrir le panneau de configuration et le pilote ODBC, choisir l'option DSN Système et le bouton "Ajouter". Nommer la source de données, par exemple, quinquin, et indiquer le chemin qui mène à sa localisation sur le disque dur, C:\base\quincaill.mdb.

2) pour chaque requête, constituer un fichier IDC, que l'on peut, soit réaliser à partir d'un assistant sous Access, soit directement avec un éditeur de texte pur. Pour les requêtes SQL, les fichiers IDC sont de la

forme suivante , par exemple pour SQL3 , le fichier sql3.idc sera :

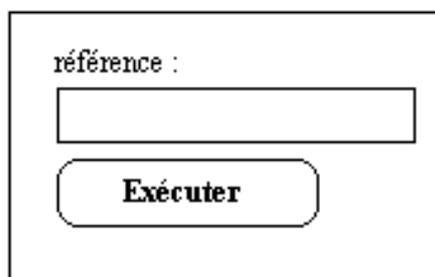
```
Datasource:quinquin
Template:lscat.htx
SQLStatement:SELECT Scategorie.nomscat
+FROM Scategorie
+WHERE (((Scategorie.nomcat)='%cat%'));
Password:
Username:
```

Datasource est le nom de la source de données donnée dans le pilote ODBC ; Template est le nom du fichier modèle HTX qui donnera les réponses ; SQLStatement est la requête SQL (donnée avec précision comme on peut le remarquer) ; le paramètre est écrit dans une syntaxe spéciale : '%cat%' ; Password et Username sont utilisés pour le contrôle des droits d'accès.

3) Les requêtes sont provoquées par l'utilisateur, généralement par un clic souris. Ainsi, pour l'écran 3, on aura dans le corps du document HTML correspondant :

```
<form action = " scripts/sql3.idc " method = " GET ">
référence : <br>
<input type = " text " size = " 20 " name = " numero "><br>
<input type = "submit " value = " Exécuter ">
</form>
```

ce qui correspond à la figure suivante :



The image shows a rectangular box representing a web form. Inside the box, the text "référence :" is displayed. Below this text is a horizontal rectangular input field. Underneath the input field is a rounded rectangular button with the text "Exécuter" centered on it.

On notera que le fichier sql3.idc de requête est placé dans un sous-répertoire " scripts " ; il faut en effet procéder ainsi et déclarer ce sous-répertoire comme exécutable dans l'administration du serveur.

**P2** : La restitution des résultats.

Ayant opté pour la méthode ODBC, la restitution des résultats s'effectuera par les fichiers modèles HTX. Prenons tout d'abord l'exemple de l'affichage de la liste des sous-catégories provoquée par la requête SQL3, et donc le fichier sql3.idc. Le fichier qui permet de renvoyer les réponses est lscat.htx. Sa syntaxe est celle de HTML comme indiquée au chapitre 4. Toutes les réponses à la requête seront écrites entre la balise spéciale <%begindetail%> et la balise spéciale <%enddetail%>, par exemple sous forme d'une table :

```
Liste sous-catégories de la catégorie : <%nomcat%> <br>
<br>
<table border="1" cellspacing="0" width="100%">
<THEAD>
<tr>
<th >sous-catégories</th>
</tr>
</THEAD>
<TBODY>
<%begindetail%>
<tr>
<td align="center" valign="top"><a href=scripts/sql4.idc><%nomscat%></a><br></td>
</tr>
<%enddetail%>
</TBODY>
<TFOOT>
</TFOOT>
</table>
```

Avec ce code HTML, les sous-catégories s'afficheront sous forme d'une liste dont chaque ligne est un lien qui renvoie vers une requête figurant ici avec le fichier sql4.idc (et correspond à la requête SQL4 définie plus haut).

Examinons maintenant l'affichage des caractéristiques d'un article suite à la requête SQL1. L'article est retrouvé dans la base de données avec la fourniture du paramètre donnant la référence de l'article. Le fichier sql1.idc a la forme suivante :

```
Datasource: quinquin
Template: article.htx
SQLStatement: SELECT article.numref, article.image,
article.libelle, article.dim1, article.dim2, article.dim3,
article.poids, article.pu, article.dispo, article.delai
+FROM article
+WHERE (((article.numref)='%param%'));
```

Password:

Username:

Le fichier de réponse article.htx affiche dans un document HTML un tableau à deux colonnes avec les caractéristiques de l'article à gauche et l'image de l'article à droite :

```
<body>
<p>article n°<%numref%> </p>
<%begindetail%>
<table border="1" width="100%">
<tr>
<td>caractéristiques : <br>
libellé : <%libelle%><br>
longueur : <%dim1%><br>
largeur : <%dim2%><br>
hauteur : <%dim3%><br>
poids : <%poids%><br>
prix unitaire : <%pu%><br>
disponibilité : <%dispo%><br>
délai de réapprovisionnement : <%delai%><br>
</td>
<td width="50%"> </td>
</tr>
</table>
<%enddetail%>
</body>
```



## 5 - Réalisation - cas n° 2 : le syndicat d'initiative

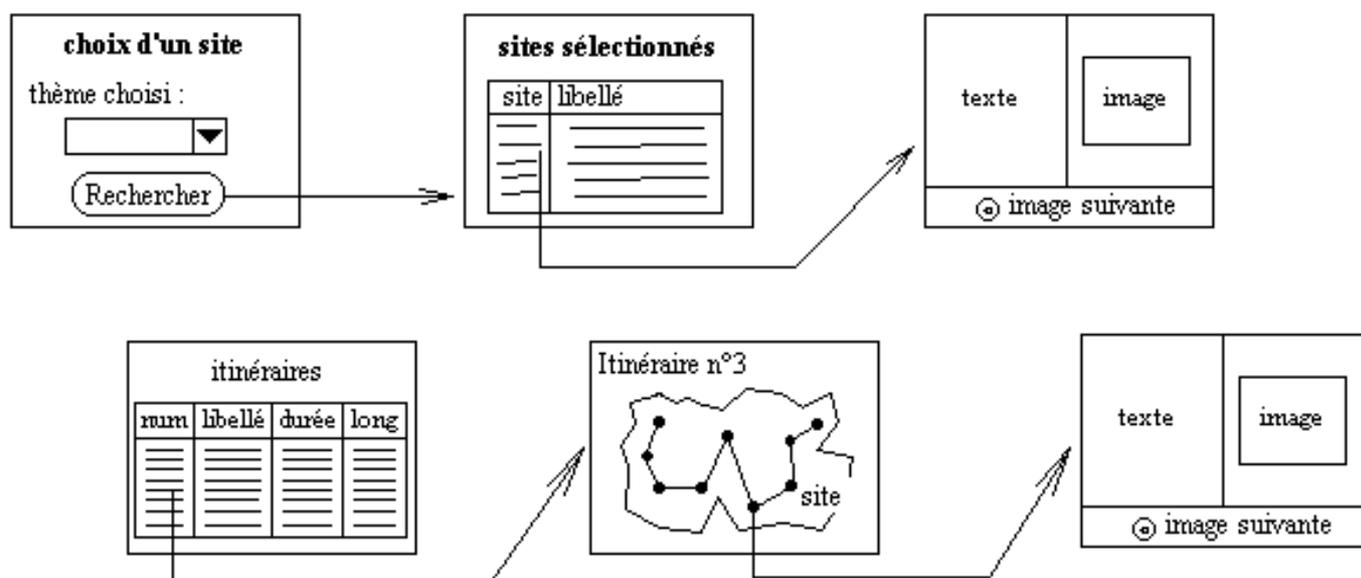


La réalisation peut être très proche de celle du cas précédent. Le serveur est basé sur un micro-ordinateur connecté à Internet (et déclaré comme serveur Web) et doté

- d'un logiciel serveur Web
- d'un SGBD relationnel du commerce
- d'un pilote ODBC

La base de données réside également sur le poste serveur ainsi que le site Web Sitot (Syndicat d'Initiative pour le Tourisme Ouvert à Tous).

Sur le poste client équipé d'un logiciel navigateur, l'utilisateur recevra du serveur des documents HTML pouvant servir pour la formulation des requêtes et pour la visualisation des résultats. Les figures qui suivent indiquent pour la recherche d'un site à partir d'un thème et pour la recherche d'un site à partir d'un itinéraire la succession des écrans que l'on peut imaginer :



Le mécanisme le plus simple est celui de l'utilisation d'ODBC. Le pilote ODBC doit être défini avec le nom de la source de données. Les requêtes sont formulées via des fichiers IDC, les réponses sont données via des fichiers HTX suivant des procédures très similaires à celles de l'étude de cas précédente.



## 6 - Réalisation - cas n° 3 :

### la médiathèque sur le patrimoine



Compte tenu des volumes à transférer, la consultation à distance par Internet devra attendre encore un peu. La consultation peut néanmoins s'effectuer dans le cadre d'un réseau local à 10 Mbits/s (Ethernet, Token Ring) ou même 100 Mbits/s (fast Ethernet). Une solution, bien répandue actuellement, peut consister en un réseau local sous Windows NT, avec postes clients sous Windows 95.

Le serveur sous Windows NT est doté d'un SGBD relationnel du commerce et les traitements de consultation peuvent être réalisés par programmation dans des langages disponibles sur ce type de plate forme.



# Gestion de l'information multimédia

## Annales d'examens

---

### Objectifs

1. examen terminal juin 2000
2. examen terminal décembre 2000
3. examen terminal juin 2001
4. examen terminal décembre 2001
5. examen terminal juin 2002
6. Examen terminal de décembre 2002
7. examen terminal de juin 2003

 [Téléchargement du chapitre \(exam.pdf - ... Ko\)](#)

---

Concepteurs : 

Gérard-Michel Cochard

## Annales d'examens

---

### Objectifs du chapitre :



- ◆ .....
  - ◆ .....
  - ◆ .....
  - ◆ .....
  - ◆ .....
- 



Heure chez vous :

# Examen terminal de juin 2000

---

- Avertissement :**
- 1) l'usage de documents est autorisé**
  - 2) la durée de l'épreuve est de 2 heures**
- 

La commune de Vasy possède un patrimoine historique important qui remonte au Moyen-Age où en 1245, le comte Jean de Carcassonne battit à plates coutures le roi Cédric de Bourgogne, événement resté célèbre sous le nom de bataille de Vasy. Durant les siècles, de nombreux événements eurent également lieu dans ce site, par ailleurs très pittoresque.

Le maire de Vasy a soigneusement collectionné des gravures, dessins, photos, illustrations cartes postales,...ayant trait à divers événements sur sa commune. Il souhaite, pour en conserver la mémoire, mais aussi pour en assurer une consultation aisée, numériser tous ces documents que l'on supposera ici tous du type image.

Le nombre d'images rassemblées par le maire, Monsieur Alain Térieur, est d'environ 10000 et la taille moyenne d'une image, une fois numérisée dans un format courant BMP est de 40 Ko. Par ailleurs, sa collection grandit chaque jour, car ses adjoints, l'instituteur, le curé, les habitants lui apportent d'autres documents (environ 1000 par an).

Ne sachant comment organiser sa collection et ne disposant d'aucun matériel informatique le maire fait appel à un étudiant du DESS SIM ou du DU Techniques de Base pour le Multimédia pour effectuer la conception et la réalisation d'un système d'information multimédia permettant

- d'archiver les images numérisées
- de consulter l'ensemble des images suivant divers critères de recherche :
  - par date ou grande période de l'histoire
  - par thème ( architecture, métiers anciens, fêtes et jeux , ...)
  - par événement (la bataille de Vasy, l'inauguration de la salle des fêtes en 1922, la libération, ....)
  - par personnage célèbre (le roi Cédric de Bourgogne, le président Félix Faure, le général De Gaulle, ....)

Le conseil municipal, sur la proposition de son maire, a voté un budget de 50 KF pour cette étude. Pour la réalisation, une fois chiffrée (ce n'est pas l'objet de cette épreuve), une demande de soutien sera effectuée auprès du Conseil Général et du Conseil Régional.

Il vous est demandé de brosser rapidement

- Le schéma relationnel de la base de données, support du système d'information
  - Les technologies d'acquisition, de traitement, de stockage, d'accès à mettre en œuvre.
-

## Examen terminal de décembre 2000

---

- Avertissement :**
- 1) l'usage de documents est autorisé
  - 2) la durée de l'épreuve est de 2 heures
- 

Une communauté de communes avait un projet de réalisation d'un CD-ROM pour faire la promotion de son territoire et disposer d'un instrument de communication.

Un avant-projet a été établi pour la conception et la réalisation de ce CD-ROM. Cet avant-projet a été validé par les commanditaires. Il figure en **Annexe**.

Cependant, entre temps, la communauté de communes a changé d'avis sur le support et désire maintenant disposer d'un site Web avec les fonctionnalités identiques à celles qui étaient prévues dans le CD-ROM, et, peut être aussi, des fonctionnalités supplémentaires permises par les facilités de communication d'Internet.

Il vous est demandé de faire une proposition dans laquelle on se bornera à indiquer

- les modifications à apporter au projet initial (tant sur la conception que sur la réalisation)
  - les fonctionnalités nouvelles à prendre en considération.
-

## **Atelier de Développement Multimedia**

**Avant-Projet**

**Syndicat Mixte du Pays de  
Somme Sud Ouest  
CINQ CANTONS**

## Objectif

Le projet consiste en la conception et la réalisation d'un CD-ROM permettant de présenter le Pays de Somme Sud Ouest pour en faire la promotion. Il s'agira donc de permettre la découverte des atouts que possède cette région, à savoir notamment :

- les lieux et sites touristiques
- les lieux de loisir
- les possibilités d'implantation d'entreprise
- le vade-mecum de la région

Il pourra aussi présenter l'action du Syndicat Mixte du Pays de Somme Sud Ouest (SMPSSO).

Le CD-ROM pourra être distribué

- soit par le SMPSSO dans chaque ville principale
- soit par des organismes touristiques ou de promotion
- soit par les entreprises sponsors pour leur personnel ou leurs visiteurs.

Le public est donc très large.

Il appartiendra aux commanditaires du projet de définir avec leurs propres critères le mode de diffusion du produit. L'Atelier de Développement Multimédia ( Université de Picardie Jules Verne) se chargeant de la réalisation technique du prototype.

Le développement pourra se faire à partir d'une équipe mixte composée

- de délégués du SMPSSO
- des ressources informatiques et humaines de l'Atelier de Développement Multimédia (DEP- Université de Picardie Jules Verne).
- et de toute personne susceptible d'apporter son concours dans la limite des orientations et des caractéristiques du produit.

Le CD-ROM est destiné à être utilisé sur une plateforme multimédia consistant en un micro-ordinateur de type Mac ou PC, doté d'un lecteur de CD-ROM et d'une carte son. Une capacité mémoire suffisante est nécessaire.

Le présent document est un outil de travail destiné à initialiser le processus de conception. Il est évidemment destiné à être largement amendé par les partenaires du projet.

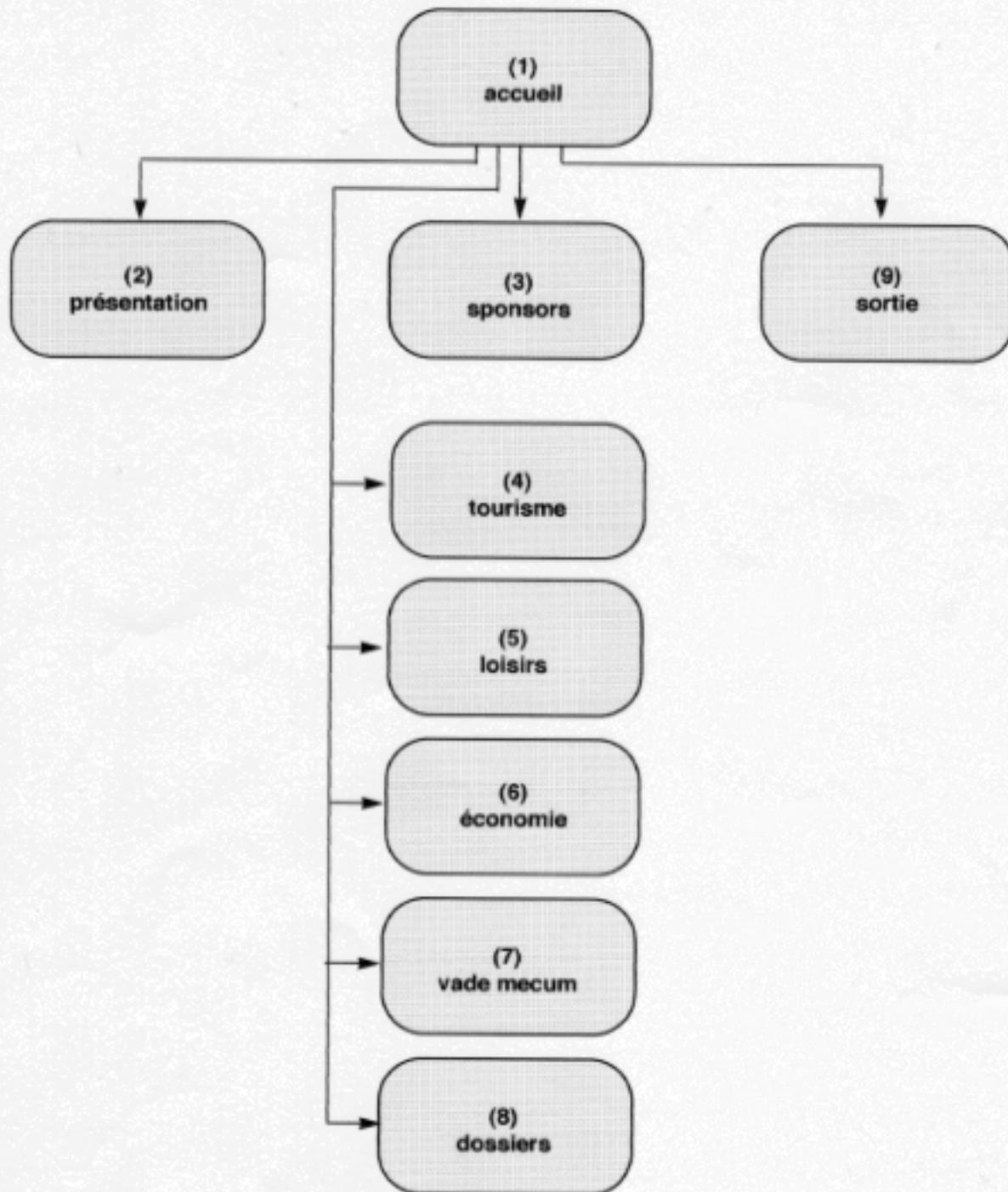
Il conviendra également de préciser au plus tôt la définition de la propriété du produit.

# Structure générale du produit

On trouvera ci-dessous un descriptif navigationnel du produit. Les écrans sont présentés de manière fonctionnelle et ne correspondent pas à leur réalisation définitive.

## Architecture globale

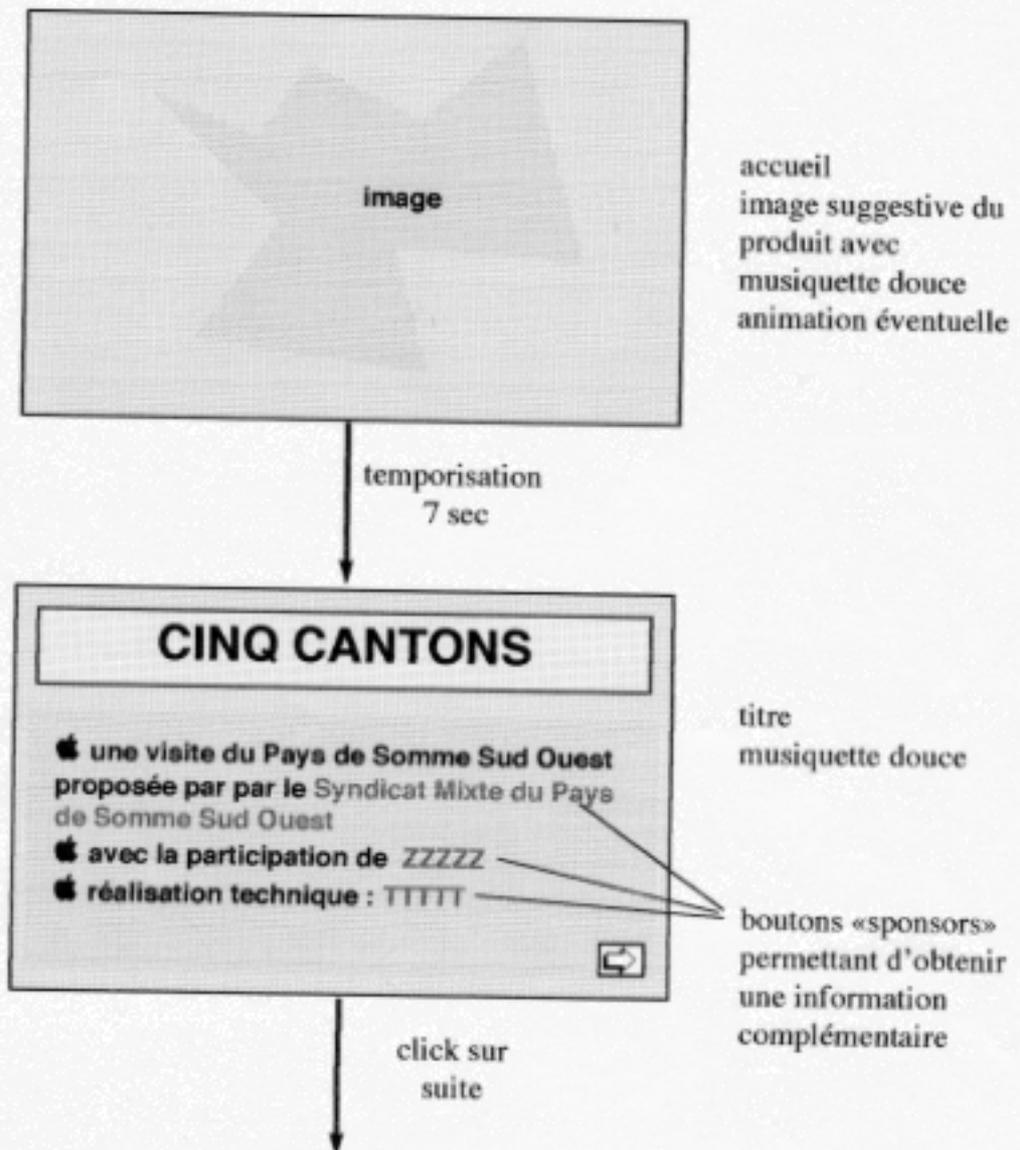
Le schéma qui suit précise la structure d'ensemble du produit en termes de navigation.

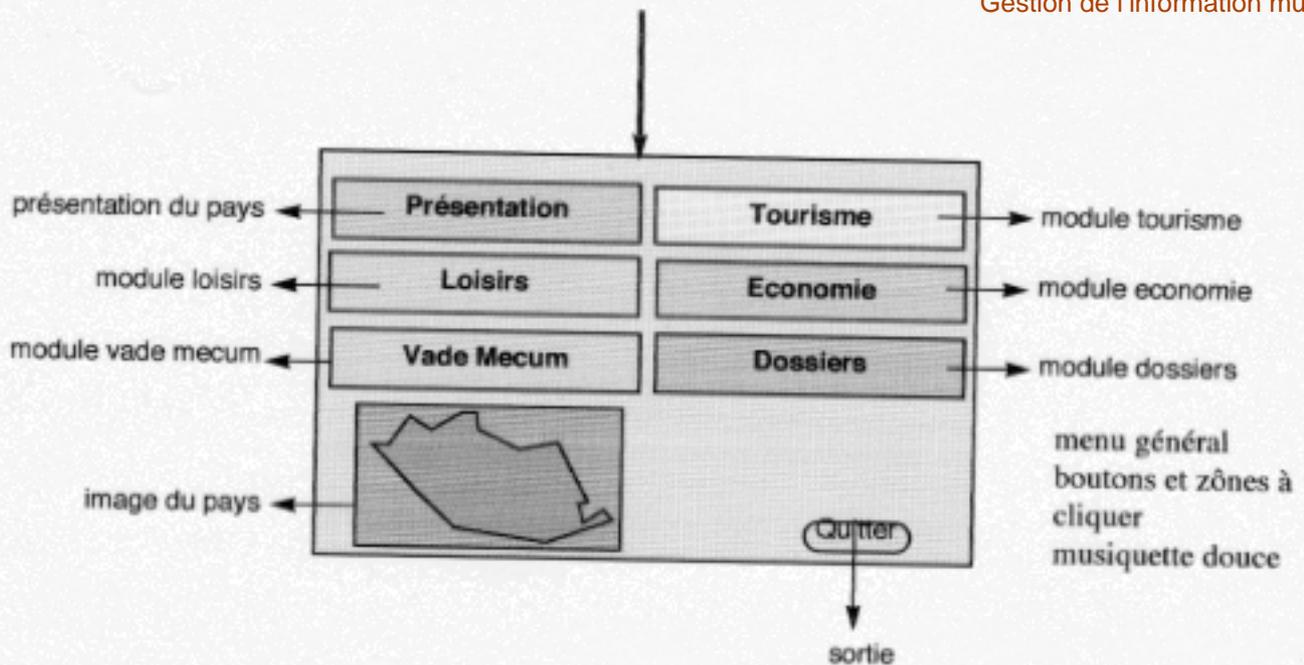


**(1) accueil**

Ce module comportera quelques écrans, notamment

- un écran d'accueil
- un écran de titre ( sujet, auteurs, partenaires) avec un bouton «sponsors»
- un écran de menu général proposant les options suivantes sous forme de boutons ou de zones à cliquer :
  - présentation du produit et de la région
  - vers module tourisme
  - vers module loisirs
  - vers module économie
  - vers module vade mecum
  - vers module dossiers
  - bouton «quitter»



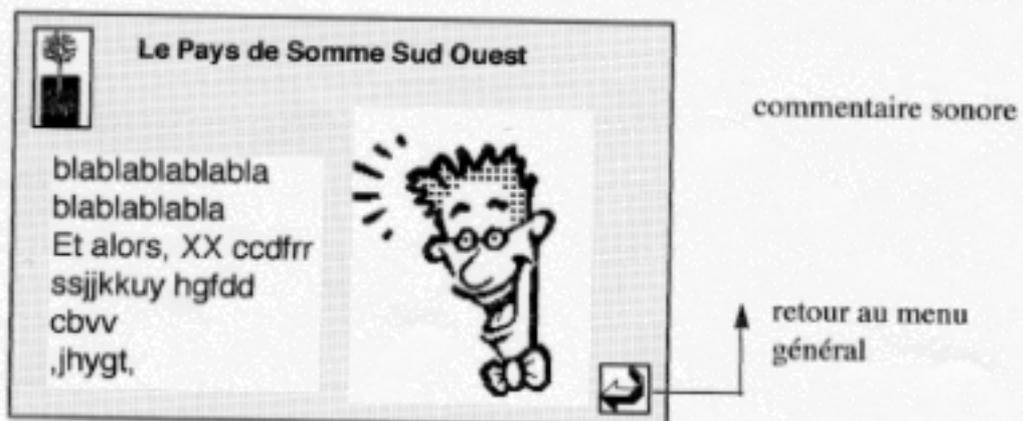


## (2) présentation

Ce court module vise à présenter le produit et sa finalité : qui, quoi, comment.

On pourrait envisager un écran avec une animation d'une durée de 2 minutes environ ; cette animation, qui doit pouvoir être interrompue à tout moment devrait présenter l'intérêt du produit en citant les passages les plus intéressants (avec extraits).

On pourrait imaginer un conteur parlant de la région des cinq cantons avec évocation géographique, historique, administrative.



**(3) sponsors**

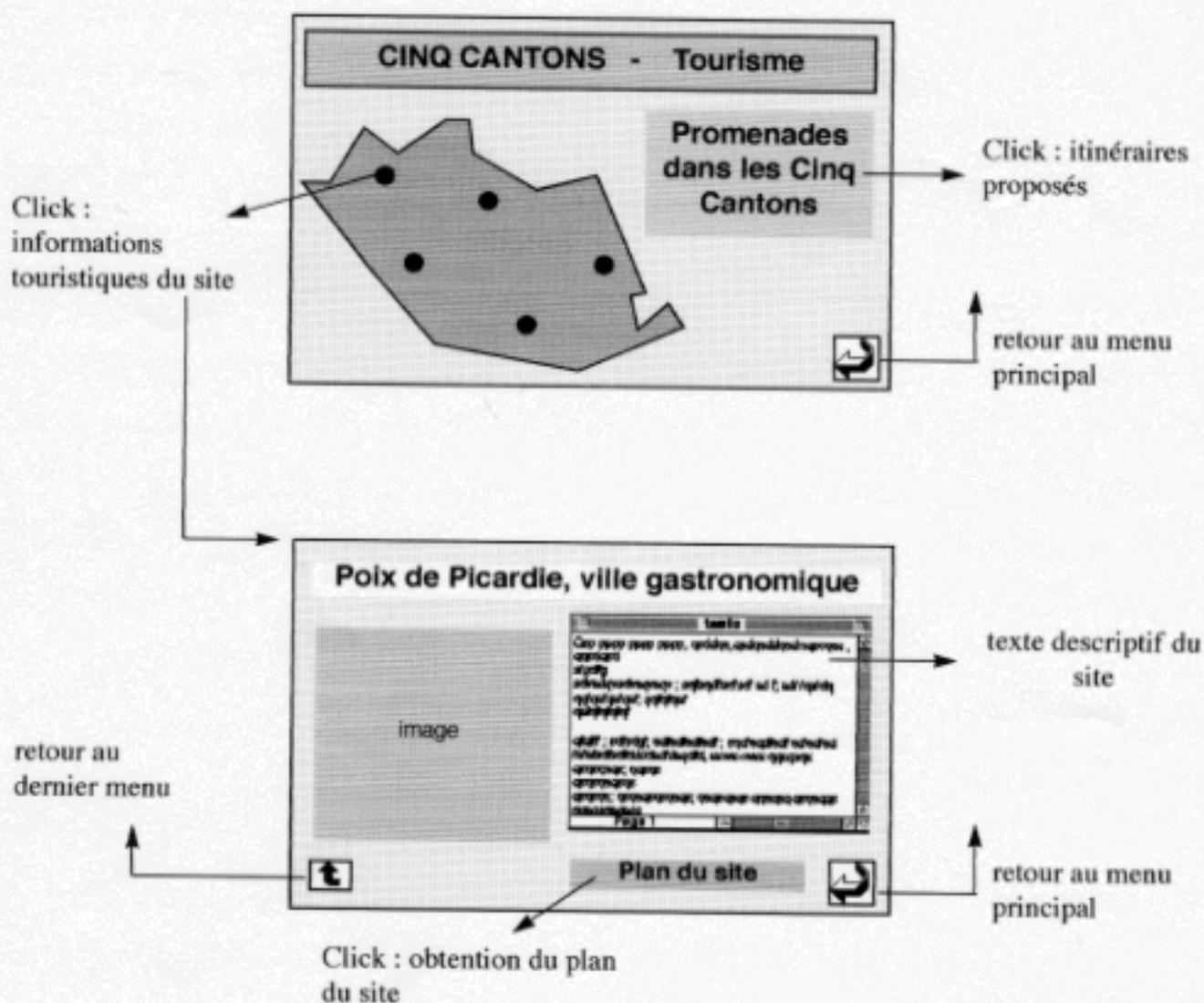
Ce module est destiné à faire connaître les sponsors du produit, c'est à dire les organismes ou entreprises ayant apporté leur participation active ou financière pour la réalisation du CD-ROM.

L'action de la souris dans le module accueil doit permettre l'apparition d'un écran donnant des informations promotionnelles sur un sponsor.

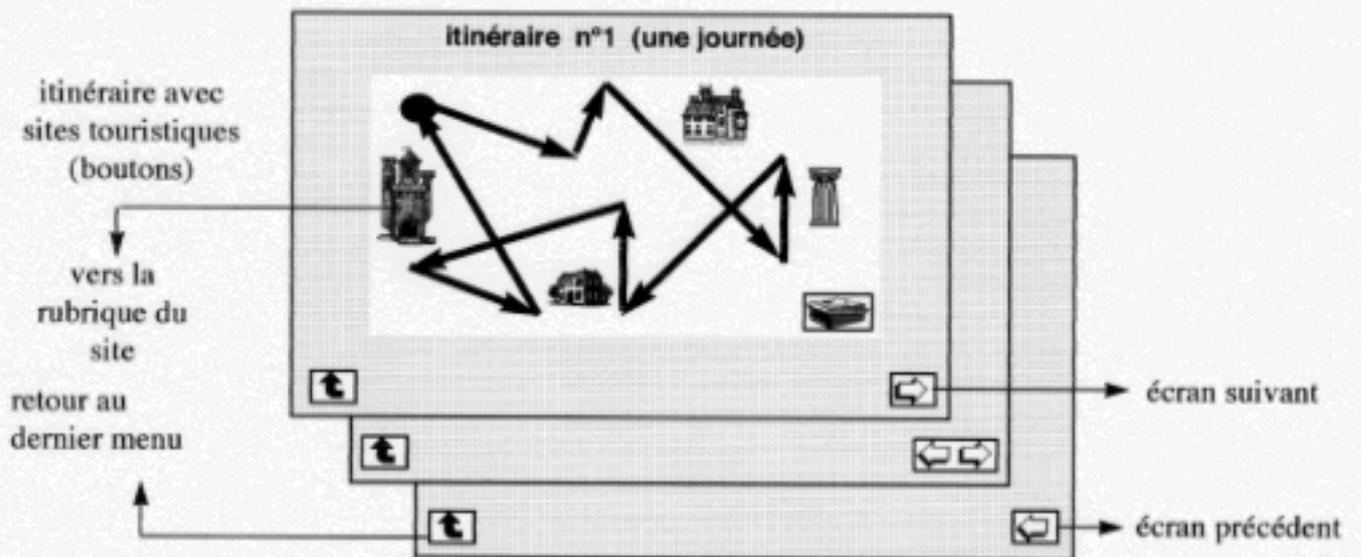
**(4) module tourisme**

Ce module est censé renseigner l'utilisateur sur les différentes possibilités touristiques de la région des Cinq Cantons. On peut envisager deux navigations de base :

- Choisir un site sur la carte et obtenir les informations touristiques en cliquant sur un lieu précis.
- Etudier quelques itinéraires touristiques pré-établis.



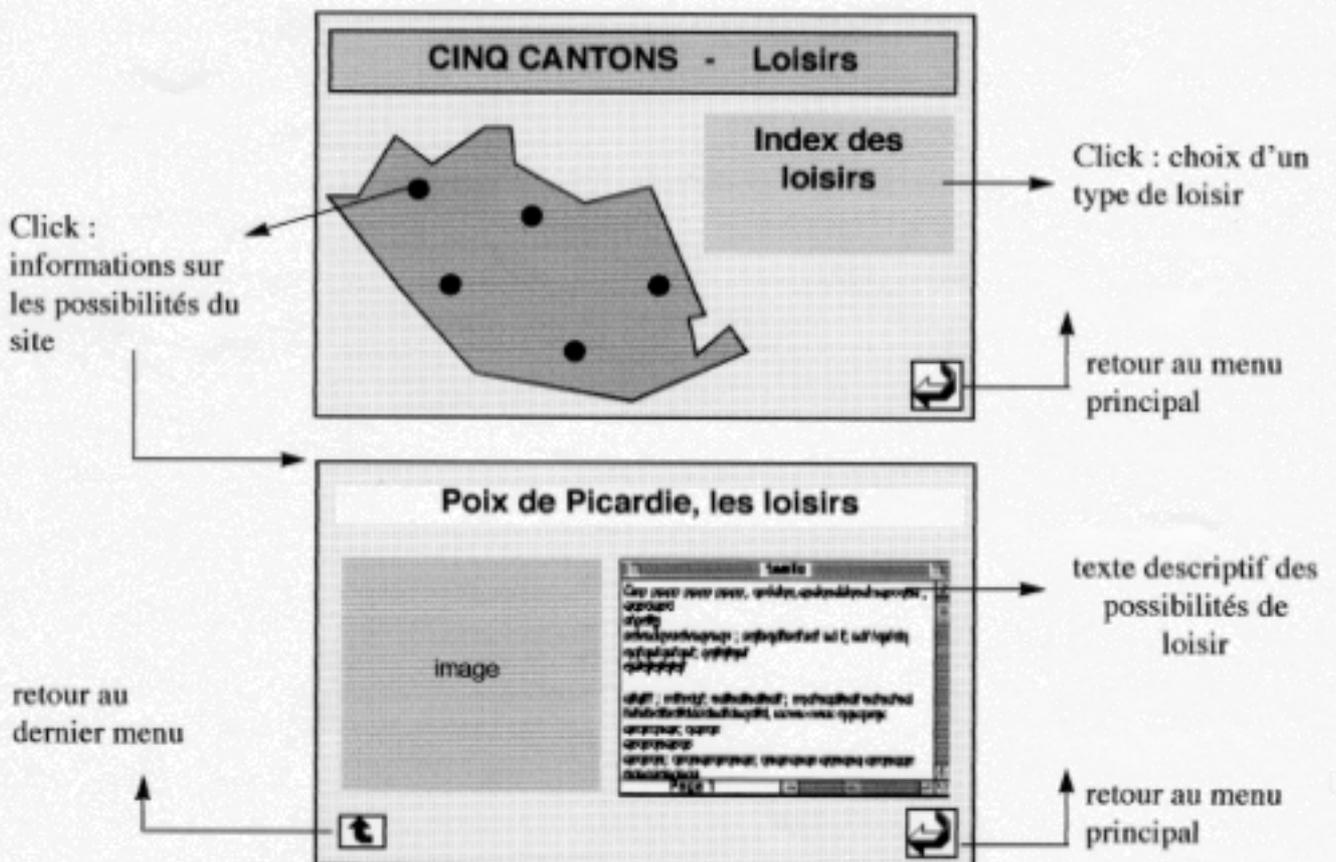
itinéraires touristiques proposés :

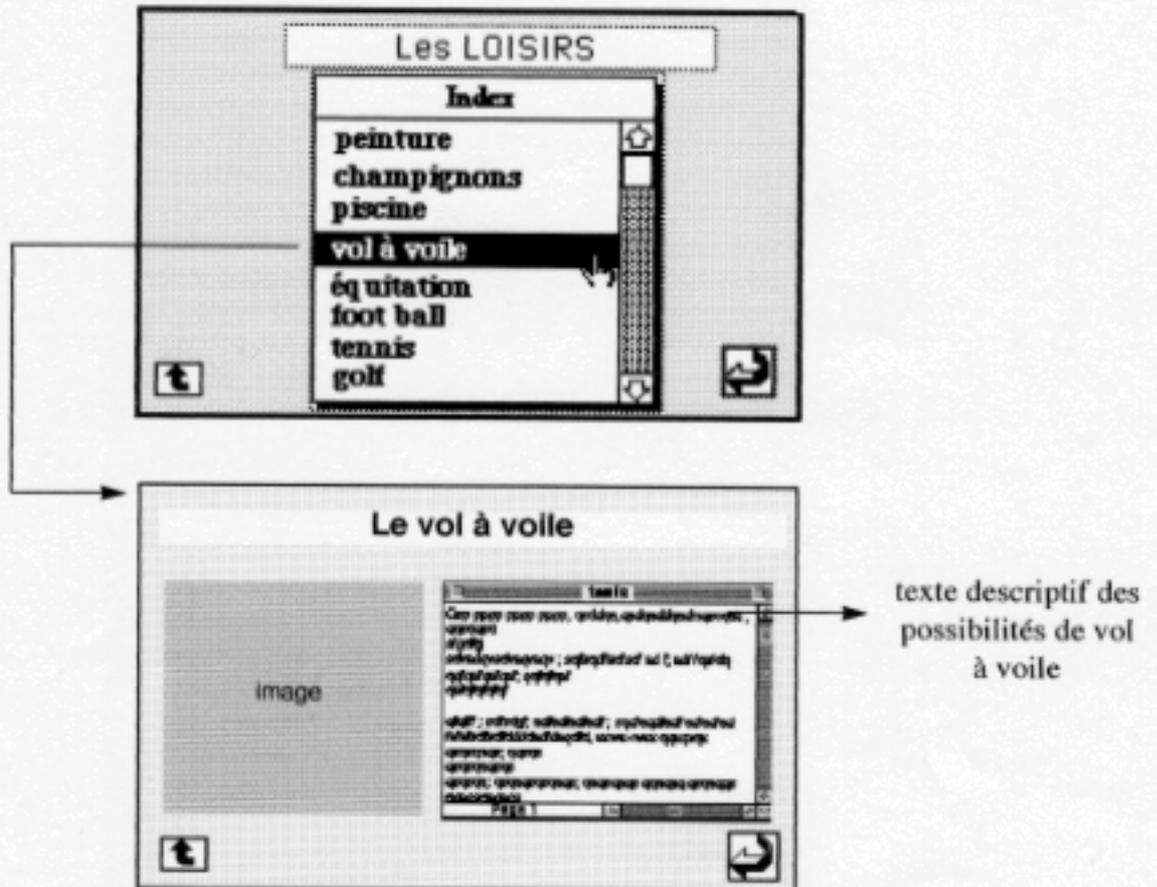


**(5) module loisirs**

La navigation peut être conçue de la manière suivante :

- ou bien choix sur la carte d'un site et affichage des possibilités de loisirs de ce site.
- ou bien choix dans un index d'un type de loisir et affichage des possibilités correspondantes.

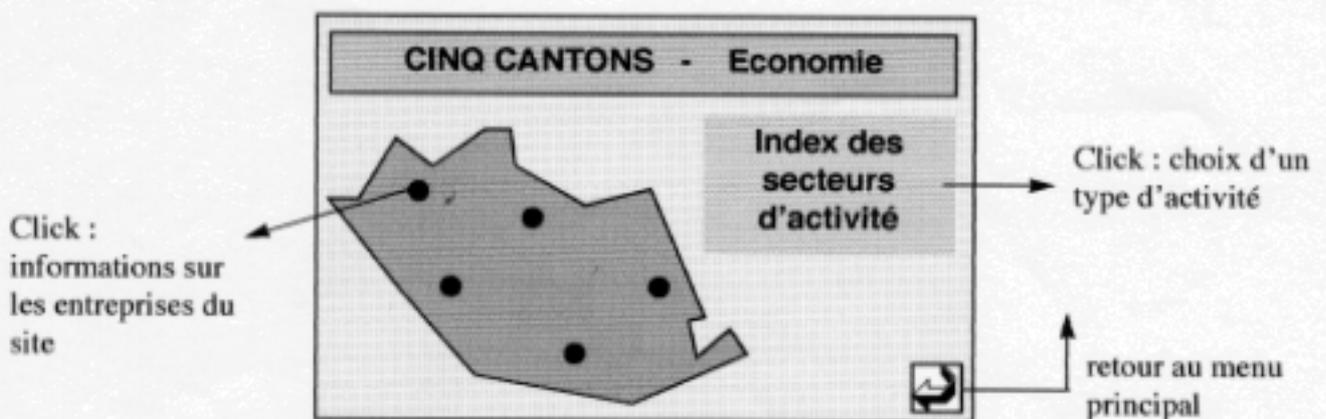


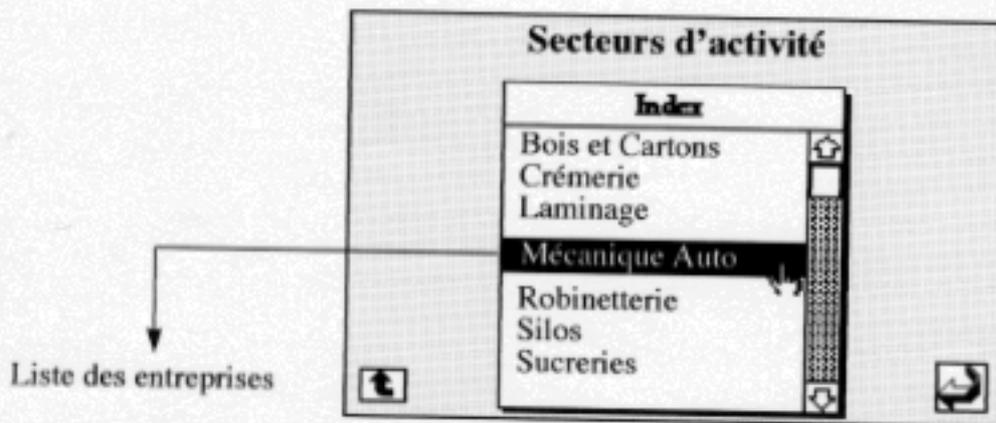


## (6) module économie

Le module économie peut être abordé

- soit à partir de la carte des cinq cantons ;
  - On obtiendra alors pour un site donné
    - la liste des entreprises implantées sur ce site
    - les conditions d'implantation des entreprises
- soit à partir d'un index des secteurs d'activité qui donnera pour un secteur donné la liste des entreprises correspondantes.



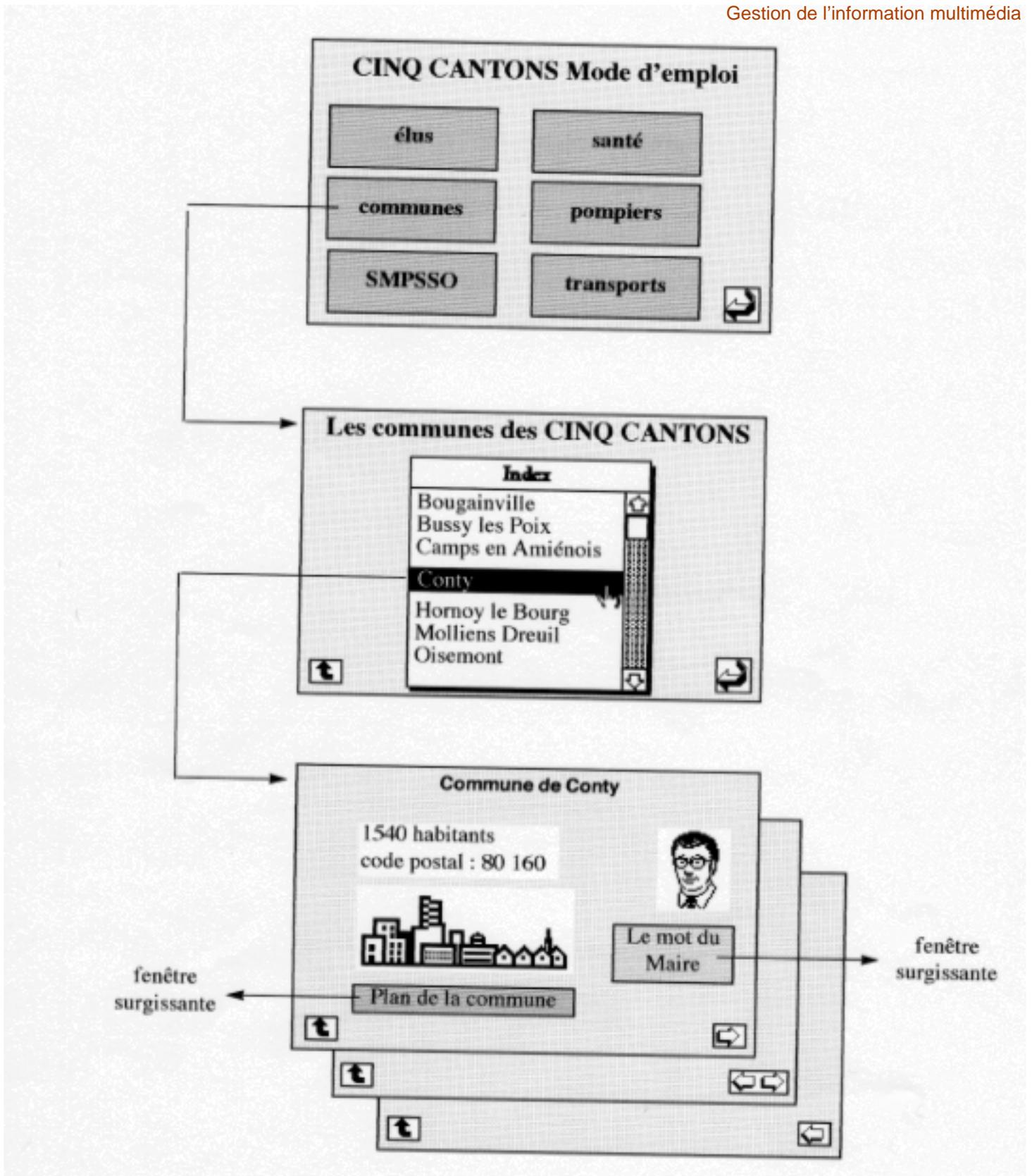


### (7) module vade mecum

Ce module est destiné à donner des renseignements utiles sur le pays des cinq cantons :

- conseillers généraux, députés, ...
- communes et conseils municipaux
- Syndicat Mixte du Pays de Somme Sud Ouest
- etc .....

Il appartiendra aux commanditaires du produit de définir avec précision le type de renseignements qu'il convient de présenter.



## **(8) module dossiers**

Ce module permet de présenter l'action du Syndicat Mixte du Pays de Somme Sud Ouest sous forme de dossiers :

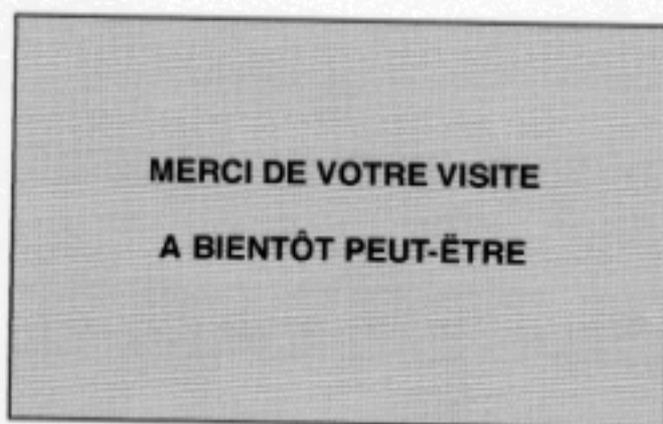
- les grandes lignes du plan triennal
- le Programme Local de l'Habitat
- les foires dans les Cinq Cantons
- etc ....

Il appartiendra aux commanditaires du produit de préciser les contenus de ces divers dossiers. On pourra toutefois s'inspirer largement du journal d'information «Les Infos du Pays de Somme».

## **(9) Sortie**

Un écran d'adieu permet de saluer l'utilisateur :

temporisation  
2 sec



## **Notes importantes**

(1) Il est rappelé que les schémas précédents ont une utilité essentiellement fonctionnelle ; ils ne préfigurent pas les écrans réels qui seront confectionnés ; d'ailleurs, ils ne sont qu'un échantillon des possibilités de navigation qui n'ont pas toutes été explorées ici.

(2) Il y a évidemment la possibilité de l'hypertexte : en cliquant sur un mot, on peut être dirigé vers des écrans sans rapport direct avec la rubrique courante. En particulier certaines rubriques pourront servir plusieurs fois dans la mesure où elles sont accessibles de diverses façons.

(3) Ce document est un document de travail qui doit être largement discuté. Toute suggestion, toute amélioration, toute critique sont bien évidemment les bienvenues. Il faut «itérer» le processus de création par des apports divers.

# Examen terminal de juin 2001

---

- Avertissement :**
- 1) l'usage de documents est autorisé
  - 2) la durée de l'épreuve est de 2 heures
- 

La ville de Vazyléon possède un riche passé historique et depuis le début du siècle, de nombreux personnages célèbres sont nés ou ont habités dans cette localité.

La municipalité a décidé de mettre en valeur, dans un but promotionnel et touristique, ces personnages célèbres dont la trace se manifeste particulièrement dans de très nombreuses photographies ou dessins . Dans cet objectif, la municipalité de Vazyléon a demandé à l'un de ses habitants, par ailleurs étudiant en DESS SIM, de concevoir une base de données multimédia présentant les personnages, leurs photos, et divers documents de type texte ou image.

La conception de la base de données n'est pas demandée ici. Cependant, il est clair que l'on aura à gérer le stockage et l'accès à un ensemble d'images. On peut évaluer le fond « image » à environ 5000 images de taille moyenne 500 Ko en format JPG. Toutefois, ces documents n'existent, au départ, que sous forme « papier ».

Par ailleurs, il est demandé à ce que l'on puisse accéder à cet ensemble d'informations depuis un poste client connecté à Internet.

Le travail demandé consiste à apporter des solutions à l'acquisition, au stockage et à l'accès depuis Internet à cette base de données textes-images.

---



## Institut Universitaire Professionnalisé MIA GE

DESS Systèmes d'Information Multimédias

Diplôme d'Université Techniques de Base pour le Multimédia

EXAMEN

Code du Module : T5

Intitulé du module : Gestion de l'information multimédia

Nom de l'enseignant : Gérard-Michel Cochard

Session de décembre 2001

Durée de l'épreuve : 2 heures

NB : les documents sont autorisés

Une université souhaite mettre en oeuvre un site Web destiné :

- A l'information du public externe (étudiants cherchant une formation, entreprises cherchant des partenaires en recherche/développement, ... ) ;

- A l'information interne (communication entre les différents usagers universitaires).

1) Quels types d'information pouvez vous identifier pour la construction de ce site Web ?

2) Quelles structures de stockage et de gestion pouvez-vous proposer ?

3) Quels dispositifs de présentation des informations pouvez-vous proposer ?

4) Quels formats de présentation des données pouvez-vous proposer ?



## Institut Universitaire Professionnalisé MIAGE

DESS Systèmes d'Information Multimédias

Diplôme d'Université Technique de Base pour le Multimédia

EXAMEN

Code du Module : T5

Intitulé du module : Gestion de l'information multimédia

Session de juin 2002

Nom de l'enseignant : Gérard-Michel Cochard

Durée de l'épreuve : 2 heures

NB : l'utilisation de documents, de calechettes, d'ordinateurs non connectés est autorisée.

1) On désire afficher sur un écran, par l'intermédiaire d'un navigateur très récent (par exemple IE6), le document suivant (le logo provient du fichier image logo.gif) :



**Diplôme d'Université**  
**"Techniques de Base pour le Multimédia"**

Liste des modules :

- T1 : Image informatique**
- T2 : Réseaux**
- T3 : Approches orientées objet**
- T4 : Interfaces**
- T5 : gestion de l'information multimédia**

1a) Comment, en HTML, s'écrirait ce document ?

1b) On désire utiliser XML pour décrire la structure de ce

document. Faire une proposition.

1c) Quelle DTD pourrait accompagner le fichier XML du 1b) ?

1d) Pour l'affichage du document XML, on utilise une feuille de style CSS. Faire une proposition de feuille CSS.

1e) Alternativement, on utilise XSL pour l'affichage du document XML. Faire une proposition de fichier XSL associé.

---

2) L'Université de Picardie Jules Verne, au travers de ses filières professionnelles, impose une période de stage en entreprise à ses étudiants. Ceux-ci, à l'issue de leur période en entreprise, doivent rendre un rapport de stage dont la structure est la suivante :

- Présentation de l'entreprise d'accueil
- Présentation de la mission confiée à l'étudiant
- Rapport sur le déroulement de la mission
- Présentation des résultats de la mission
- Conclusions

Un rapport de stage contient du texte et des images exclusivement.

Ces rapports sont encore aujourd'hui réalisés sous forme papier ce qui rend difficiles leur archivage et leur consultation. Il est envisagé de numériser les rapports de stage de manière systématique.

Il vous est demandé de faire des propositions sur la démarche à suivre pour aboutir à cette numérisation. Les aspects

- édition des rapports,
- archivage des rapports,
- consultation sélective de parties de rapports
- infrastructures matérielles et logicielles

seront étudiés



Institut Universitaire Professionnalisé MIAGE

DESS Systèmes d'Information Multimédias  
DU Techniques de base du multimédia

EXAMEN

Session de décembre 2002

Code du Module : T5

Intitulé du module : Gestion de l'information Multimédia

Nom de l'enseignant : Gérard-Michel Cochard

Durée de l'épreuve : 2 heures

- 1) On désire afficher sur un écran, par l'intermédiaire d'un navigateur très récent (par exemple IE6), le document suivant concernant une fiche signalétique d'étudiant (la « photo » provient du fichier image photo.gif) :

<p><b>LHERMITTE Bernard</b></p> <p><b>promotion 24</b></p> <p><b>Modules validés :</b></p> <p><b>T1 : juin 2002</b></p> <p><b>T2 : juin 2002</b></p> <p><b>T4 : juin 2002</b></p> <p><b>T3 : non validé</b></p> <p><b>T5 : décembre 2002</b></p>	
--	---

- 1a) Comment, en HTML, s'écrirait ce document ?
- 1b) On désire utiliser XML pour décrire la structure de ce document. Faire une proposition.
- 1c) Quelle DTD pourrait accompagner le fichier XML du 1b) ?
- 1d) On utilise XSL pour l'affichage du document XML. Faire une proposition de fichier XSL associé.
- 2) Le réseau des services de formation continue des universités souhaite mettre en œuvre un site d'information à destination des publics salariés des entreprises ainsi qu'aux demandeurs d'emploi. Ce site devrait présenter

- des renseignements généraux : qu'est-ce que la formation continue universitaire, les différentes formules de formation (cours du soir, formation à distance, alternance, ...), la procédure de validation des acquis de l'expérience, les événements et manifestations significatifs
- un annuaire des services de formation continue comportant pour chacun d'eux une fiche signalétique de chaque service : coordonnées générales, coordonnées de la personne de contact, université de rattachement et logo, thèmes principaux de formation, services d'appui. Un écran spécial devrait promouvoir à des moments donnés des actions spécifiques du service concerné.

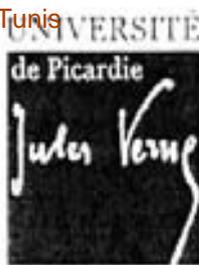
On part de l'hypothèse que tous les services ont accès au réseau Renater (Internet des universités). Le site d'information se trouve sur un serveur géré par un des services.

On demande d'établir un avant-projet faisant clairement apparaître

- l'organisation générale du dispositif envisagé (description des composants du site)
- les contraintes techniques
- les outils mis à disposition de l'utilisateur
- les procédures de mise à jour

Comme il s'agit d'un avant-projet, il n'est pas demandé de détailler trop profondément les items ci-dessus, mais plutôt de faire apparaître les problématiques qui vous semblent les plus pertinentes.

NB : l'utilisation de documents, de calculettes, d'ordinateurs non connectés est autorisée.



Institut Universitaire Professionnalisé **MIAGE**  
DESS Systèmes d'Information Multimédias  
DU Techniques de Base du Multimédia

EXAMEN  
Session de juin 2003

Code du Module : T5  
Intitulé du module : Gestion de projets multimédias  
Nom de l'enseignant : Gérard-Michel Cochard

Durée de l'épreuve : 2 heures

Note : l'usage de documents est autorisé

### 1<sup>ère</sup> partie

La chaîne internationale d'hôtels Biendormir comporte 236 établissements répartis dans 57 pays et situés dans des lieux touristiques. Une action de promotion vise à proposer sur un site Internet implémenté sur un seul serveur Web, de la manière la plus vivante et attractive, une présentation de chacun des établissements de la chaîne Biendormir.

Pour chaque établissement, on prévoit

- une description générale avec une petite visite 3D de l'hôtel
- une description de l'environnement (monuments, sites à visiter, plans de ville, ....)
- une description des activités proposées

On suppose que vous êtes une société de développement multimédia souhaitant proposer ses services à la chaîne Biendormir.

- 1) Y a-t-il des aspects bases de données multimédias dans ce projet et si oui préciser ces aspects (on pourra donner quelques exemples de tables et/ou d'enregistrements) ?
- 2) Pourriez vous donner un ordre de grandeur de la capacité de stockage nécessaire au serveur Web ?

2<sup>ème</sup> partie

Une école comprend plusieurs classes. Une classe comprend des élèves. Chaque élève est décrit par son nom, son prénom et une appréciation globale numérique de 0 à 20. Un exemple d'occurrences figure ci-dessous :

classe	nom	prénom	appréciation
M1	DURAND	Jacques	9
	DUPONT	Joséphine	12
	DUGENOU	Gédéon	18
	DUFOUR	Gaétan	6
M2	DUBOIS	Mireille	8
	DUCHEMIN	Micheline	11
	DURACUIRE	Caroline	13
	DUPIPEAU	Berthe	4
M3	DUBONNET	Jules	10
	DUBALAI	Jim	17
	DUDUCHE	Joseph	16
	DUMONT	Marthe	12
	DUVAL	Charles	9

- 1) Réaliser un fichier XML correspondant à ces données
- 2) Donner une DTD accompagnant ce fichier XML
- 3) Ecrire un document XSL permettant d'afficher les données sous forme du tableau précédent.
- 4) Idem en n'affichant dans le tableau que les élèves ayant une appréciation inférieure à 10.

## Bibliographie

J.C. HANKE HTML 4 XML Micro Application

P. CHALEAT, D. CHARNAY Programmation HTML et JavaScript Eyrolles

J. TERRASSON Les outils du multimédia éditions Armand Colin, 1992

J-M. HERELLIER Le Multimédia Sybex

T. VAUGHAN Multimedia, making it work Mc Graw Hill

E. HOLSINGER

Le Multimédia...Comment ça  
marche ?Dunod

A. FLORY, F. LAFOREST

Les bases de données  
relationnelles

Economica

G. GARDARIN Bases de données Eyrolles

S. REYMANN Informatique DECF, MSTCF (\*) Foucher

S. MIRANDA, J-M. BUSTA

L'art des bases de données ;

Introduction aux bases de  
données

Eyrolles

S. MIRANDA, J-M. BUSTA

L'art des bases de données ;

Les bases de données  
relationnelles

Eyrolles

C.J. DATE

An Introduction to Database  
Systems

Addison-Wesley

G. GARDARIN, P. VALDURIEZ SGBD avancés Eyrolles

G. GARDARIN

Internet/Intranet et bases de  
données

Eyrolles

A. MESGUICH, B. NORMIER

Comprendre les bases de  
données( $\mu$ )

Masson

G. GARDARIN, P. VALDURIEZ

Bases de données  
relationnelles ; analyse et  
comparaison des systèmes  
Eyrolles

G. GARDARIN

Bases de données ; objet et  
relationnel  
Eyrolles

D. NANJI, B. ESPINASSE, B.  
COHEN, H. HECKENROTH

Ingénierie des systèmes  
d'information avec Merise ;  
vers une deuxième génération  
Sybex

N. KETTANI, D. MIGNET, P.  
PARE, C. ROSENTHALSABROUX  
De Merise à UML Eyrolles  
(\* ) pour débutants

Etudes :Outils pour la téléconsultation  
et la distribution sur réseau de  
documents audiovisuels et multimédias

1. Stockage numérique de masse

INA

2. Serveurs "média"

3. Systèmes de compression

4.I. Réseaux et transmission  
multimédia

4.II. Réseaux et commerce  
électronique

G. DUPOIRIER Technologie de la GED Hermes

S. MIRANDA, J-M. BUSTA

L'art des bases de données,  
tome 1 : Introduction aux bases  
de données

Eyrolles

R.M. KAPLAN Intelligent Multimedia Systems Wiley

R. BERNARD L'Intranet en entreprise Sybex