

DESS Sigma : Module 653
Cartographie et SIG interactifs en ligne
Séance 3 : Approfondissement de l'utilisation de MapServer

Après avoir installé un serveur SIG MapServer sous Windows avec le paquet MS4W de MapTools, nous allons explorer et en illustrer par des exemples les possibilités offertes par les mapfiles et le fonctionnement avec les templates (fichiers modèles) HTML.

Préambule : Les diverses manières d'utiliser MapServer.

MapServer est à l'origine une application à la ligne de commande, qui a donné naissance à plusieurs versions, offrant les mêmes possibilités de cartographie mais dans des contextes différents.

MapServer CGI : C'est le mapserver.exe de la version Windows, c'est à dire une application brute. Cette version est une application CGI, c'est à dire accessible à distance dans une installation serveur web. CGI signifie Common Gateway Interface, c'est un protocole utilisé par tous les serveurs et navigateurs pour permettre l'accès distant à des applications (sous certaines conditions de sécurité). Dans ce cas, on fournit les paramètres dans l'url, au minimum un mapfile et un mode par exemple :

<http://localhost/cgi-bin/mapserv.exe?map=../htdocs/raster/rastr1.map&mode=map>

MapServer MapScript : C'est la version « bibliothèque de fonctions » du logiciel. Elle est disponible pour les langages PHP, Python, Perl, Java. MapServer est alors disponible sous la forme de fonctions accessibles directement dans le langage de programmation. Par exemple, en PHP, on charge la bibliothèque de fonctions (library), puis on peut ouvrir ou générer un mapfile, modifier des paramètres, faire des sélections, et générer des cartes, etc. dans le script PHP.

1- Description illustrée de la structure des mapfiles.

Documentation officielle : <http://www.mapserver.org/mapfile/>

Les paramètres obligatoires ou importants à fournir dans un mapfile sont repérés par un soulignement.

Les mapfiles sont des fichiers texte qui vont contenir tous les paramètres nécessaires à MapServer pour la génération d'un document cartographique, statique ou dynamique. Ils sont organisés en blocs (ou objets) hiérarchiques, imbriqués, notés NOM_DU_BLOC ... END. Le bloc racine est le bloc MAP (Tout mapfile est donc compris entre une balise MAP et une balise END).

Généralités

Les mapfiles ne sont pas sensibles à la casse (minuscules / majuscules), sauf pour les noms des champs attributaires, notés entre [crochets], et naturellement les valeurs de type chaîne, notées entre guillemets ou apostrophes.

Un mapfile contient au maximum 50 couches (layers) (mais on peut modifier cette valeur dans le code source et recompiler MapServer).

Les chemins peuvent être indiqués de manière absolue (à partir de la racine du système), ou relativement à l'emplacement du mapfile. Les chemins vers les données sont relatifs au paramètre SHAPEPATH (en l'absence de ce dernier les données seront recherchées dans le même répertoire que le mapfile).

Le **bloc MAP** : Ce bloc est la racine du mapfile, il va contenir les paramètres concernant la globalité de la carte à générer.

Paramètres généraux

NAME : nom de la carte, préfixe ajouté à tous les fichiers image générés par le mapfile, donc à garder de courte taille, et entre guillemets.

UNITS : Unité de la carte, utilisé pour les calculs et le dessin de la barre d'échelle. L'information sur l'unité du fond de carte n'est pas toujours apportée par le bloc « projection », il faut donc essayer de la fournir explicitement à chaque fois.

Valeurs : [feet|inches|kilometers|meters|miles|dd]

EXTENT : Extension de la carte, donc coordonnées des extrémités de la carte dessinée dans l'image générée par le mapfile.

Valeurs : [xmin] [ymin] [xmax] [ymax]

STATUS : On ou Off, active ou désactive la totalité du mapfile.

FONTSET et **SYMBOLSET** : chemin et nom des fichiers contenant les définitions de polices de caractère (utile sous linux) et de symboles (on peut aussi définir un symbole directement dans le mapfile, mais c'est utile de pouvoir les rassembler dans un seul fichier qui pourra être appelé par différents mapfiles).

SHAPEPATH : Chemin vers le répertoire contenant les fichiers de données (des couches), ou le chemin de base à partir duquel les paramètres DATA iront les chercher.

ANGLE : Angle de rotation de l'image, en degrés. Utilisé en mode MapScript, demande un bloc PROJECTION.

DEBUG : Active le débogage sur fichier de log et en précise le niveau. Une variable système "MS_ERRORFILE" doit être présente et indiquer le chemin du fichier de log.

Valeurs : [off|on|0|1|2|3|4|5]

Paramètres de génération de l'image

IMAGETYPE : type de fichier image à générer. Correspond à l'ensemble inclut par défaut de blocs OUTPUTFORMAT. Ce paramètre dépend des options avec lesquelles à été compilé MapServer.

Valeurs : [gif|png|jpeg|wbmp|gtiff|swf|user_defined]

SIZE : Dimensions (largeur et hauteur) de l'image à générer, en pixels, séparées par un espace.

RÉSOLUTION : Résolution de l'image en dpi, par défaut 72. N'affecte que les calculs d'échelle, cf. le bloc SCALEBAR.

IMAGECOLOR : Couleur de fond de l'image de la carte (RGB).

Le bloc MAP va contenir, en plus des paramètres généraux, les blocs de second niveau, qui vont permettre de spécifier le contenu de la carte.

le bloc WEB

Ce bloc va paramétrer le fonctionnement du serveur Web autour de la carte générée par MapServer.

IMAGEPATH : Chemin absolu vers le répertoire temporaire où sera stocké le fichier image généré par le mapfile. Doit se terminer par un slash « / », de préférence utiliser des slashes, pas des anti-slashes « \ » (même sous windows).

IMAGEURL : Base de l'url (partie de l'adresse web s'ajoutant au nom du serveur dans l'url) pointant sur le répertoire temporaire contenant les fichiers image générés. Correspond à l'alias défini dans la configuration du serveur Web.

EMPTY : Url vers laquelle l'utilisateur est redirigé lorsqu'une requête MapServer ne retourne pas de résultats.

ERROR : Url vers laquelle l'utilisateur est redirigé lorsque se produit une erreur.

LOG : Chemin et nom du fichier texte dans lequel sera inscrite l'activité de MapServer sur ce mapfile. Contient une trace des opérations effectuées et des éventuelles erreurs survenus. Fonctionne en parallèle avec le paramètre DEBUG et les paramètres LOG des blocs LAYER du mapfile.

MINSCALE : Échelle minimale à laquelle le mapfile sera dessiné. Si une échelle plus petite est demandée, MapServer dessinera la carte à l'échelle précisée par ce paramètre.

MAXSCALE : Idem pour l'échelle maximale.

TEMPLATE : Chemin et nom du fichier modèle HTML éventuel (cf. seconde partie du présent support de cours, dédiée à ce sujet).

HEADER : Modèle HTML à utiliser avant l'insertion de l'image de la carte.

FOOTER : Idem pour le bas de page.

METADATA : Bloc secondaire utiliser pour stocker des paires nom – valeur. Utilisé par les modèles HTML (pour stocker des variables généralisées) et en mode serveur WMS/WFS.

Le bloc **REFERENCE**

Ce bloc définit les paramètres de la petite carte utilisée comme référence pour la carte principale. C'est une image sur laquelle va être dessiné un rectangle représentant l'extension de la carte principale, ou la localisation des résultats d'une requête, en mode QUERY.

Paradoxalement, MapServer a besoin d'une image fixe représentant la petite carte de référence, il ne va pas la générer. Cela permet par contre l'utilisation d'une image externe. Il faut donc alors, pour produire cette image, utiliser un mapfile simplifié (uniquement les contours de la couche principale par exemple), réglé pour générer une image de petites dimensions. MapServer va faire la relation entre la carte principale et la petite carte de référence grâce aux paramètres EXTENT des deux cartes.

Par défaut le bloc REFERENCE a un paramètre STATUS à la valeur Off, pour activer cette carte il faut donc penser à rajouter « STATUS ON » dans le bloc.

Le bloc **LEGEND**

MapServer peut générer trois types de légendes :

- légendes simples sous forme d'images ;
- légendes basées sur un modèle de légende HTML ("template", voir ci-après) ;
- légendes HTML pur.

Les légendes simples sont des images, incluses ou pas dans l'image de la carte principale, reprenant chaque classe nommée des layers du mapfile et son figuré. Il faut donc penser à nommer toutes les classes que l'on veut voir apparaître en légende (paramètre NAME). Lorsque la légende simple est incluse dans l'image de la carte (paramètre : STATUS EMBED), on peut préciser l'endroit où la légende sera dessinée avec le paramètre POSITION. Ce paramètre prend une valeur correspondant à un code composé de deux lettres, la première pour le haut / bas, u pour « upper », l pour « lower », la seconde pour gauche / droite, l pour « left », r pour « right ».

Les caissons (rectangles) de légende pour les couches de polygones sont réglables en taille avec le paramètre KEYSIZE (valeurs : largeur, espace, hauteur) et leur espacement avec le paramètre KEYSPPACING (valeurs : écart horizontal espace écart vertical). Il n'est pas possible de titrer une légende directement (tout comme pour la carte, on peut par contre le faire en insérant la légende dans une page html ou utiliser un template html).

Le bloc **PROJECTION**

L'indication d'un géoréférencement pour la carte produite par le mapfile nécessite un bloc PROJECTION. Ce bloc contient les informations de référencement, qui peuvent être de

deux formes, une série de paramètres PROJ.4 (Cf. http://proj.maptools.org/gen_parms.html) ou un code EPSG (Cf : <http://www.epsg.org/>).

Par exemple :

```
PROJECTION
"proj=utm"
"ellps=GRS80"
"zone=15"
"north"
"no_defs"
END
```

Ou :

```
PROJECTION
"init=epsg:28992"
END
```

Le bloc **SCALEBAR**

MapServer gère les échelles selon une technique assez particulière. En effet, il part du principe que la carte sera au final une image possédant des dimensions en pixels, qui sera visualisée au moyen d'un écran qui possède une certaine résolution. Par ailleurs, la carte doit être dessinée dans un rectangle d'extension maximale donné par le paramètre EXTENT. L'échelle finale de l'image doit donc être définie selon ces paramètres. Le paramètre EXTENT prime sur le paramètre SCALE, car c'est lui qui définit plus précisément ce que doit contenir la carte à dessiner.

Pour dessiner une échelle indiquant une certaine longueur terrain, il faut donc déterminer combien de pixels cette longueur va représenter (ou procéder à des essais), et l'indiquer dans le paramètre SIZE du bloc.

Le bloc SCALEBAR permet de dessiner des barres d'échelle, dans l'image de la carte ou comme une image distincte. Il possède les paramètres suivants :

POSITION : Code à deux lettres définissant l'endroit où sera dessinée l'échelle, ce code est le même que celui utilisé par le paramètre POSITION du bloc LEGEND, cf. ci-dessus.

SIZE : Dimensions en pixels (largeur espace hauteur) du rectangle contenant la barre d'échelle. Important car détermine la longueur totale de la barre.

INTERVALS : Nombre de subdivisions à afficher.

STATUS : Inclusion (EMBED), dans une image à part (ON) ou annulation (OFF).

STYLE : Apparence de la barre, 0 donnant une barre de rectangles pleins de couleurs alternées, 1 une barre fine munie de repères (barbules vers le haut).

UNITS : Unités pour le calcul de la longueur des intervalles et l'affichage de l'unité de la barre d'échelle. Toutes unités possibles sauf degrés décimaux.

IMAGECOLOR : Couleur RGB du rectangle qui contient l'échelle.

BACKGROUNDCOLOR : Couleur RGB de la barre d'échelle et de ses libellés.

COLOR : Couleur alternative à BACKGROUNDCOLOR si barre de type 0 et plusieurs intervalles spécifiés.

OUTLINECOLOR : Couleur RGB de la réserve autour de la barre d'échelle (mais pas autour des libellés).

TRANSPARENT : Valeur booléenne (ON / OFF) qui précise si le rectangle contenant l'échelle est transparent.

Le bloc **OUTPUTFORMAT**

Ce bloc permet de définir précisément le format d'image du fichier qui sera généré par MapServer. Le paramètre général IMAGETYPE correspond en fait à des blocs OUTPUTFORMAT prédéfinis dans MapServer, par défaut. On peut ainsi mieux préciser certains paramètres de la sortie image, par exemple le taux de compression pour le format JPEG.

Exemple pour produire un JPEG en compression peu destructive :

```
OUTPUTFORMAT
NAME jpegfull
DRIVER "GD/JPEG"
MIMETYPE "image/jpeg"
IMAGEMODE RGB
EXTENSION "jpg"
QUALITY=100
END
```

Exemple pour utiliser la sortie avec anticrénelage AGG sur du PNG transparent :

```
OUTPUTFORMAT
NAME 'AGGA'
DRIVER AGG/PNG
IMAGEMODE RGBA
END
```

Le bloc LAYER

Ce bloc va définir les propriétés de création et d'affichage d'une couche de données SIG par MapServer. Les blocs LAYER sont dessinés dans l'ordre du mapfile, c'est à dire que le premier bloc du mapfile est dessiné en premier, les suivants viendront par-dessus sur la carte. L'ordre du mapfile est donc l'ordre inverse de la superposition verticale des sources.

Paramètres généraux :

NAME : Nom de la couche, utilisé comme identifiant par l'interface web. Doit être unique dans le mapfile et d'une longueur maximale de 20 caractères. Entre guillemets de préférence (ou apostrophes).

GROUP : Groupe auquel le LAYER appartient. Utilisé dans les modèles HTML pour activer/désactiver les couches par groupes.

METADATA : Bloc secondaire utiliser pour stocker des paires nom – valeur. Utilisé par les modèles HTML et en mode serveur WMS.

STATUS : Statut (visibilité) du layer. Valeurs : default, on, off. Doit prendre la valeur « default » pour que le layer soit visible lorsque l'on utilise MapServer en mode Map (le STATUS ON ne suffit pas à le rendre visible, il faut que le layer soit expressément requis).

TYPE : Type d'objet géométrique ou modalité selon laquelle la couche doit être dessinée. Valeurs : point|line|polygon|circle|annotation|raster|query. Ce paramètre peut prendre une valeur différente du type géométrique des objets contenus dans la couche d'origine, par exemple une couche de polygones peut être représentée comme un LAYER de type POINT, ce qui affichera les centroïdes des polygones (

MINSCALE : Échelle minimale à laquelle la couche sera dessinée. Si une échelle plus petite est demandée, MapServer dessinera la couche à l'échelle précisée par ce paramètre.

MAXSCALE : **Idem pour l'échelle maximale.**

SYMBOLSCALE : Échelle à laquelle les symboles et/ou les textes apparaissent à leur taille normale. Ce paramètre permet un dimensionnement dynamique de ce type d'objets selon l'échelle de la carte, dans les limites des deux paramètres précédents. Obligatoire pour l'utilisation du paramètre SIZEITEM dans un bloc CLASS.

OPACITY : Degré de transparence de la couche, exprimé en pourcentage (sans le signe %), de 100 – opaque à 0 – totalement transparent.

OFFSITE : Le numéro d'index de la couleur d'une couche raster à traiter comme transparent. Cela permet de ne garder que la région utile d'une couche raster.

POSTLABELCACHE : Valeur booléenne (true / false) qui indique à MapServer de dessiner cette couche après avoir dessiné les libellés (labels) qui sont dans le tampon des libellés. Prend la valeur false par défaut.

CLASSITEM : Nom de la colonne attributaire qui est utilisée dans les expressions de sélection des blocs CLASS.

LABELITEM : Nom de la colonne attributaire qui fournira le texte des étiquettes.

TEMPLATE : Nom du fichier modèle HTML qui prend en compte cette couche. Obligatoire pour rendre cette couche interrogeable par requête, même si on n'utilise pas de modèle HTML.

DEBUG : Valeur On ou Off. Si le paramètre général LOG est défini, les messages de débogage détaillés seront ajoutés au fichier de log, en plus des messages d'erreur.

Paramètres de données :

- **Shapefiles**

Les shapefiles sont la source de données la plus courante pour MapServer, qui au départ était construit uniquement autour de ce format (issu de la bibliothèque shapelib). On indique que la source de données du layer est un shapefile avec le paramètre DATA, qui prend comme valeur le chemin vers le shapefile, relatif au chemin SHAPEPATH (préciser l'extension .shp est inutile). Le paramètre FILTER permet d'utiliser les expressions régulières pour filtrer les objets de la source de données au niveau du bloc LAYER, c'est à dire avant le traitement des blocs CLASS qui peuvent eux aussi comporter une sélection.

L'utilisation d'une source de données shapefile offre de plus la possibilité de lier (jointure) une table dbf de données attributaires, en utilisant un bloc secondaire JOIN. Ces données deviennent ainsi interrogeables par MapServer en mode Query.

- **Couches vectorielles accédées avec OGR**

La bibliothèque de fonctions OGR permet à MapServer de lire un grand nombre de formats SIG vectoriels : http://www.gdal.org/ogr/ogr_formats.html

Pour indiquer une source de données OGR, il faut utiliser la syntaxe suivante :

```
CONNECTIONTYPE OGR  
CONNECTION [nom de la source]
```

Ce dernier paramètre dépend du type de source, généralement il s'agit du chemin relatif au SHAPEPATH et du nom du fichier, ou du nom du répertoire (source ArcInfo coverage par exemple). Certains types de sources vectorielles sont organisées en couches multiples par fichier, il faut donc choisir la couche à utiliser avec le paramètre DATA [numéro/nom de la couche]. MapServer peut aussi récupérer en partie les éventuels styles d'affichage présents dans les couches, avec le paramètre STYLEITEM AUTO. Le paramètre FILTER est utilisable aussi avec ce type de données, comme pour les shapefiles.

- **Source serveurs de données**

MapServer peut se connecter à des serveurs SIG directement, et ainsi assembler des couches de diverses origines de façon transparente. Cela permet de construire une organisation des données dispersée sur des serveurs spécialisés. Les paramètres de connexion sont fournis dans une chaîne de caractères passée en valeur au paramètre CONNECTION. Le paramètre FILTER peut contenir une expression de requête SQL (c'est à dire le texte venant après un élément SQL WHERE).

ArcSDE :

```
CONNECTIONTYPE sde  
CONNECTION "myhost,esri_sde,gisdb,userid,password"  
DATA table, colonne_geom
```

PostGIS (objet de la prochaine séance) :

```
CONNECTIONTYPE postgis  
CONNECTION "user=nom_utilisateur dbname=nom_bdd host=serveur"  
DATA "colonne_géométrique FROM table_ou_requête_SQL"
```

Oracle Spatial :

```
CONNECTIONTYPE oraclespatial  
CONNECTION "snom_utilisateur/mot_de_passe@bdd"  
DATA "colonne_géométrique FROM table_ou_requête_SQL"
```

Web Map Services (WMS)

L'utilisation d'une source WMS nécessite la présence d'un bloc PROJECTION dans le bloc MAP, ainsi que l'utilisation d'un bloc METADATA (dans le bloc WEB et dans les blocs LAYER) qui va préciser la requête faite au serveur. Cette fonctionnalité sera explorée en détails lors d'une prochaine séance.

• Couches raster

Les données raster géoréférencées sont gérées avec le paramètre TYPE raster, le paramètre DATA devant pointer vers le fichier (et son éventuel chemin relatif à SHAPEPATH). GDAL supporte un grand nombre de formats de fichier, cf. : http://www.gdal.org/formats_list.html

MapServer possède quelques fonctions de traitement des rasters, accessibles avec le paramètre PROCESSING. GDAL est capable de réaliser des traitements de rééchantillonnage / mise à l'échelle (SCALE), de tramage (DITHERING), et de sélection de bande (BAND). Cf. : la documentation spécifique aux données raster : http://mapserver.gis.umn.edu/docs/howto/raster_data

Les sources de données raster composées de nombreux fichiers juxtaposés sont gérables en une seule fois par MapServer, sous la forme d'un tuilage (tiles). Ce tuilage consiste en un fichier shape qui contient des objets polygone représentant l'extension de chaque tuile raster, et ayant comme attribut le nom du fichier. Ce type de tuilage peut être produit en utilisant l'utilitaire gdalindex distribué avec GDAL. Cela permet une gestion beaucoup plus rapide des grandes couvertures raster, notamment si les bitmaps sont eux aussi optimisés pour être lus à différentes résolutions (index pyramidal dans les tif par exemple).

Le bloc CLASS

Ce bloc permet de définir des classes thématiques dans la couche, qui vont pouvoir être affichées différemment sur la carte globale. Les blocs CLASS sont traités dans l'ordre du fichier map, selon l'ordre de classement vertical. Ce bloc peut contenir les paramètres suivants :

NAME : Nom de la classe, à préciser si l'on veut trouver cette classe dans la légende.

EXPRESSION : Critère de sélection des objets de la couche qui vont être inclus dans la classe en cours. Comme vu lors de la deuxième séance, ces sélections peuvent utiliser quatre méthodes : comparaison de chaînes de caractères (en utilisant CLASSITEM), comparaisons logiques simples (avec opérateurs), expressions régulières, fonction de chaîne length().

COLOR : Couleur de fond des objets possédant une surface, exprimée en RGB, trois valeurs entières séparées par des espaces.

OUTLINECOLOR : Couleur de contour des objets, en RGB aussi.

SYMBOL : Numéro ou nom du symbole, qui doit être défini dans le mapfile par un bloc SYMBOL ou dans un fichier SYMBOLSET lié au mapfile.

SIZE : Taille du symbole ou de la trame, uniquement utilisable avec les symboles redimensionnables.

MINSIZE et **MAXSIZE** : tailles mini et maxi (en pixels) de dessin des symboles. En dehors de cette fourchette les symboles sont dessinés à la valeur la plus proche.

SYMBOLSCALE : Échelle à laquelle les symboles et/ou les textes apparaissent à leur taille normale. Obligatoire dans le cas de symboles proportionnels avec le paramètre SIZEITEM.

TEXT : Nom de la colonne attributaire contenant le texte à utiliser pour l'étiquetage des objets de la classe (On peut aussi créer des expressions combinant plusieurs attributs entre crochets).

TEMPLATE : Chemin et nom du fichier modèle HTML éventuellement utilisé.

DEBUG : Valeur On ou Off. Si le paramètre LOGFILE est défini, les messages de débogage détaillés seront ajoutés au fichier de log, en plus des messages d'erreur.

Les bloc CLASS peut contenir des blocs secondaires LABEL et STYLE.

Le bloc LABEL

Ce bloc permet de configurer l'étiquetage des éléments de la classe, selon un champ attributaire indisué par le paramètre LABELITEM du bloc LAYER qui contient le bloc LABEL.

Paramètres de base :

TYPE : Type de police de caractère à utiliser. Avec la valeur « bitmap », MapServer utilise ses propres polices internes, bitmap donc, mais elles en peuvent subir de rotation ni être dimensionnées précisément (cf. ci-après pour la taille de police). Avec la valeur « truetype », MapServer utilise une police vectorielle, au format truetype. Le paramètre FONT indique l'alias utilisé pour indiquer le fichier ttf dans le fichier FONTSET.

Par exemple, pour utiliser la police truetype Arial, il faut un fichier déclaré par le paramètre FONTSET (du bloc MAP), et ce fichier (texte) comprendra une ligne « arial C:\windows\fonts\arial.ttf » indiquant l'alias de la police (le nom utilisé dans le bloc LABEL) et l'endroit où trouver le fichier correspondant sur la machine.

COLOR : Couleur du texte de l'étiquette, en RGB.

SIZE : Taille du texte, valeur entière pour la taille en pixels des polices TrueType, ou valeur texte pour les polices bitmap de MapServer, parmi [tiny|small|medium|large|giant].

MINSIZE et **MAXSIZE** : Tailles mini est maxi de dessin des étiquettes, en pixels.

MINFEATURESIZE : Taille minimale (valeur entière en pixels) d'un objet de la couche pour qu'il soit étiqueté. Correspond à la longueur pour les objets ligne, à la surface du rectangle d'encombrement pour les objets polygone. La valeur « auto » indique à MapServer de n'étiqueter que les objets au moins aussi gros que leur étiquette.

Paramètres d'effets d'affichage du texte

(la présence du paramètre active la fonction) :

ANTIALIAS : Réduction de « l'effet d'escalier ». Alourdit l'image produite car utilise des dégradés de couleur.

OUTLINECOLOR : Couleur de la réserve autour du texte, en RGB.

OUTLINEWIDTH : Épaisseur de la réserve, en pixels.

SHADOWCOLOR et **SHADOWSIZE** : Couleur (RGB) et décalage (en pixels) pour le dessin des ombres sous les étiquettes.

BACKGROUNDCOLOR : Couleur du rectangle qui va contenir l'étiquette.

Paramètres de positionnement :

POSITION : Valeur texte correspondant à la position de l'étiquette par rapport au centre de l'objet qu'elle renseigne, selon le schéma suivant :

ul	uc	ur
cl	cc	cr
ll	lc	lr

Avec la valeur « auto » MapServer va tester les 8 positions externes pour choisir celle qui interfère le moins avec les autres étiquettes de la classe.

ANGLE : Angle en degrés par rapport à la verticale, ou « auto » pour les objets ligne, dans ce cas MapServer alignera l'étiquette sur l'objet. Auto ne fonctionne que si les libellés utilisent une police truetype (vectorielle).

OFFSET : Valeur X Y de décalage entre le coin bas-droite de l'étiquette et le centre de l'objet renseigné.

MINDISTANCE : Distance minimale (valeur entière en pixels) entre deux étiquettes du même objet.

BUFFER : Zone tampon (en pixels) autour des étiquettes, pour éviter qu'elles se touchent.

FORCE : Force le dessin des étiquettes de la classe, sans tenir compte des contraintes de proximité. Valeur : true/false (à false par défaut).

PARTIALS : Valeur true/false qui détermine si MapServer peut dessiner des étiquettes incomplètes (coupées par les bords de l'image). Vaut false par défaut.

WRAP : Précise le caractère (entre guillemets) du texte des étiquettes pour indiquer un passage à la ligne et donc produire des libellés sur plusieurs lignes.

2- Utilisation des templates HTML.

Documentation officielle : <http://mapserver.gis.umn.edu/docs/reference/templatereference>

Dans le cadre d'une utilisation sur un serveur web, les auteurs de MapServer ont ajouté une fonctionnalité au logiciel pour lui permettre de générer non seulement une image de carte, mais un document HTML complet pouvant comprendre les divers éléments (carte, échelle, légende, titres et mentions). Ce fonctionnement utilise un fichier modèle HTML, dans lequel MapServer va modifier des valeurs pour les remplacer par les éléments qu'il va produire. Ce fichier modèle HTML est ce qu'on appelle un "template" dans le vocabulaire de MapServer.

Le fonctionnement des templates est très simple : dans le mapfile on spécifie le chemin vers un fichier template HTML. Ce dernier contient le code HTML de la page dans laquelle vont se trouver les éléments produits par MapServer, ainsi que certaines balises spéciales, qui seront remplacées par le code HTML adéquat par MapServer. La balise la plus importante est la balise [img], qui sera remplacée par MapServer par le chemin vers le fichier image de la carte qu'il génère. Le passage des paramètres entre le template HTML et MapServer passe par le biais de variables CGI, donc en utilisant les formulaires HTML si l'on veut faire simple.

Exemple :

```
<!-- MapServer Template -->
<html>
<head>
<title>MapServer : Exemple de modele HTML</title>
</head>
<body>
  <form method = POST action = "/cgi-bin/mapserv.exe">
    <input type = "submit" value = "Carte !">
    <input type = "hidden" name = "map" value = "c:/ms4w/Apache/htdocs/
templates/document.map">
    <input type = "hidden" name = "map_web_imagepath" value = "c:/ms4w/tmp/
ms_tmp/">
  </form>
  <table>
    <tr>
      <td>Titre de la carte</td>
    </tr>
    <tr>
      <td></td>
    </tr>
    <tr>
      <td></td>
    </tr>
    <tr>
      <td>Échelle : 1 / [scale]</td>
    </tr>
  </table>
</body>
</html>
```

L'exemple ci-dessus est une page HTML qui comporte un formulaire, deux images et un texte. Le formulaire sert à proposer à l'utilisateur un bouton qui, lorsqu'il est activé, envoie des paramètres cachés à MapServer, qui, en retour, renvoie la même page HTML mais dans laquelle la balise [img] de la source de l'image a été remplacée par le chemin vers le fichier image qui a été généré. De même, la balise [ref] a été remplacée par l'image de la petite carte de référence, et la balise [scale] par la valeur numérique de l'échelle.

Ce fonctionnement peut être décomposé en deux parties : un fichier d'initialisation qui fournit les paramètres de base à MapServer, et un fichier template différent qui lui va être le modèle de la page HTML à générer. Ainsi l'utilisateur ne sera pas confronté à une première page incomplète (images vides).

Un fichier template peut être la solution la plus indiquée pour utiliser MapServer comme CGI (c'est à dire sans programmer), car il peut permettre de spécifier les paramètres de base sans avoir à les écrire dans une url très longue, comme on le ferait à la ligne de commande. Un fichier template minimal est ainsi à même d'initialiser MapServer, de lui fournir les chemins vers le mapfile et le fichier image à générer, ainsi qu'un cadre de dimensions précises pour y insérer la carte.

Tous les paramètres de MapServer CGI peuvent être utilisés (et donc remplacés) dans un template HTML, et l'on peut aussi faire passer à MapServer des variables personnalisées, il les remplacera dans le HTML final selon les balises du template.