

Refonte du Site web *MADONNE*

Rapport Technique
Laboratoire L3i, Université de La Rochelle

Mathieu Delalandre

10 août 2006

Table des matières

1	Introduction	3
2	Refonte du code	4
2.1	Introduction	4
2.2	Suppression <i>PHP/MySQL</i>	4
2.3	Refonte <i>CSS</i>	4
2.4	Re-déclaration <i>Flash</i>	7
2.5	Exemple d'utilisation	9
3	Gestion du contenu	10
3.1	Introduction	10
3.2	Conception de pages <i>HTML</i>	11
3.3	Base de données bibliographiques	11
4	Re-définition de la charte graphique	13
4.1	Introduction	13
4.2	Reproduction <i>CSS</i>	14
4.3	Styles alternatifs	15
5	Conclusion	20
6	Annexes	21
6.1	Feuille de style <i>CSS</i>	21
6.2	Makefile <i>BibTeX2HTML</i>	22
6.3	Switcher de styles <i>JavaScript</i>	22
6.4	Mise en oeuvre des styles alternatifs	24
	Bibliographie	26
	Table des figures	27

1 Introduction

Durant l'année 2005-2006 j'ai occupé un poste d'Ingénieur d'Étude au laboratoire *L3i*¹ dans le cadre du projet *MADONNE*². L'objectif de ce projet était de contribuer à la réalisation de systèmes d'indexation d'images de document du patrimoine. Il a groupé différents laboratoires français au sein d'un Consortium de 2003 à 2006.

Dans le cadre des fonctions associées à mon poste il m'a été demandé de m'occuper de la refonte du site web *MADONNE* existant [1]. Ce dernier était codé en *HTML* [2] exploitant des technologies *PHP/MySQL* [3] et *Flash* [4]. Il a été conçu comme outil collaboratif et proposait pour cela différentes fonctionnalités comme un forum, des possibilités de dépôt de documents, etc. Les motivations de refonte de ce site ont été diverses :

Bilan d'usage : Après une année de mise en service le bilan d'usage du site s'est révélé peu concluant. Tout d'abord de part son faible taux de consultation lié à son manque de contenu (limité à un descriptif des sous-projets et carte des laboratoires partenaires). Ensuite, et plus particulièrement, en ce qui concerne ses fonctionnalités dynamiques (forum quasi inutilisé, documents déposés inférieurs à une dizaine, etc.).

Faible sécurité : Le site a été "attaqué" durant l'année 2006. Des failles *PHP/MySQL* ont été exploitées pour supprimer le forum mettant en cause, d'un point de vue sécurité, les technologies dynamiques utilisées.

Démonstrations web : Une des principales évolutions envisagées pour ce site concernait l'ajout de démonstrations web. Suite à différents échanges avec les membres du Consortium *MADONNE* au cours de la rentrée 2005, il a été convenu que ces démonstrations seraient localement implantées sur leurs serveurs puis pointées depuis le site web *MADONNE*. Notre objectif était alors de définir une charte graphique de façon à harmoniser les différentes pages de démonstration.

Dérive *HTML* : De façon à satisfaire des critères de style, le site existant utilisait clairement une dérive de *HTML* [5]. La balise *table* était en effet utilisée pour le formatage des contours des blocs de texte des pages.

Ajout de contenu : Il nous a été demandé l'ajout de contenu au site existant comme : une page d'introduction, une page de publications, une page de liens, etc.

Bilan d'activité : Le projet *MADONNE* arrivait en "fin de vie" à la rentrée 2006. Il nous a donc été demandé de re-orienter le site en cours, défini comme une plate-forme collaborative, vers un site bilan dans le but de valoriser les actions du Consortium.

¹Laboratoire Informatique Image Interaction
http://www.univ-lr.fr/labo/l3i/site_statique

²MAsse de DONnées issues de la Numérisation du patrimoine

À la vue de ces différents aspects nous avons entrepris la refonte du site selon deux axes directeurs :

Refonte du code : Suppression de la partie *PHP/MySQL*, refonte *CSS* et re-déclaration des animations *Flash*.

Gestion du contenu : Édition de pages *HTML* et base de données bibliographiques.

Dans la suite de ce rapport nous présentons chacun de ces deux axes dans les sections suivantes (2) et (3). Dans la section (4) nous parlerons de re-définition de charte graphique. Finalement dans la section (5) nous donnerons nos conclusions et perspectives sur ce travail.

2 Refonte du code

2.1 Introduction

Dans cette section nous présentons la refonte du code effectuée sur le site *MADONNE*. Cette refonte concerne différents aspects : suppression de la partie *PHP/MySQL*, refonte *CSS* et re-déclaration des animations *Flash*. Nous aborderons chacun de ces trois aspects dans les sous-sections suivantes (2.2), (2.3) et (2.4). Finalement, dans la section (2.5) nous présenterons un exemple d'utilisation.

2.2 Suppression *PHP/MySQL*

Relativement aux aspects **bilan d'usage**, **faille sécurité** et **bilan d'activité** présentés page 3 nous avons entrepris la suppression de la partie *PHP/MySQL* du site *MADONNE*. Celle-ci s'est limitée à la suppression de la base *MySQL*, de la page forum et à la suppression des scripts *PHP* au sein des codes *HTML* (entre autres pour le login).

2.3 Refonte *CSS*

Le concept³ de Style Sheets (*SS*), ou feuilles de style, est apparu en 1996 avec la publication de la recommandation [6] par le *W3C*. Le principe des feuilles de style consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte. Elles sont appelées Cascading Style Sheets (*CSS*) car il est possible d'en définir plusieurs et que les styles peuvent être hérités en cascade.

³Introduction extraite de : <http://www.commentcamarche.net/css/cssintro.php3>

La Figure 1 suivante donne un exemple d'utilisation de feuille de style. Deux listing de code sont présentés, un en *HTML* et son équivalent en *HTML/CSS*. Ces codes permettent tout deux de définir un paragraphe de police *Times*, de couleur verte, en italique et de taille 8.

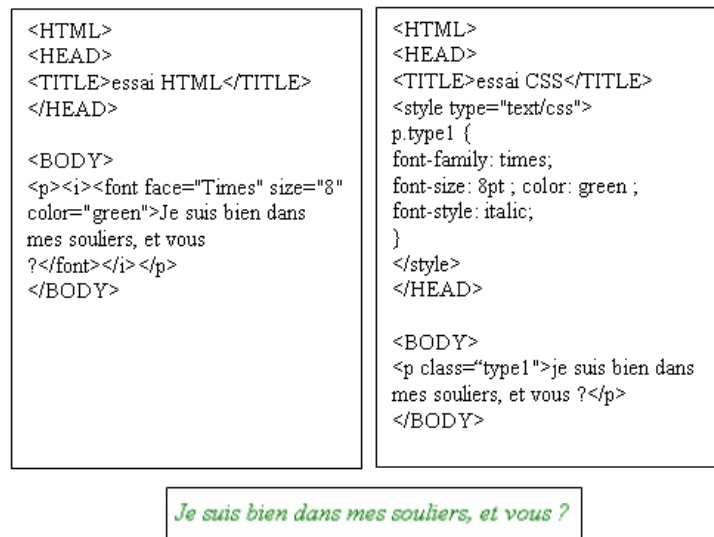


FIG. 1 – Exemple de code *HTML* vs *HTML/CSS*

Une bonne compréhension des *CSS* passe donc par une bonne maîtrise du langage *HTML*. Nous citons donc ici quelques ouvrages d'introduction disponibles sur Internet relatifs au *HTML* [7] et aux *CSS* [8] [9] [5]. En effet, les feuilles de style ont été mises au point afin de compenser les manques du langage *HTML* en ce qui concerne la mise en page et la présentation. Le *HTML* offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte. Toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent. Les feuilles de style permettent elles :

- D'obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style.
- De permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes.
- Une plus grande lisibilité du *HTML*, car les styles sont définis à part.
- Des chargements de page plus rapides, pour les mêmes raisons que précédemment.
- Un positionnement plus rigoureux des éléments.

Relativement aux aspects **démonstrations web** et **dérive HTML** présentés page 3 nous avons donc entrepris la refonte *CSS* du site web *MADONNE*. En effet, les *CSS* s'imposait comme solution pour répondre aux critères de style mis en oeuvre dans le site initial [1] et pour harmoniser les différentes pages de démonstrations des membres du Consortium. De façon à constituer notre feuille de style *MADONNE* nous avons défini, dans une première étape, quatre composants principaux : *main*, *submenu*, *title* et *block*. Ces derniers permettent de spécifier à la fois la structure logique et physique de nos pages. La Figure 2 suivante illustre leur mise en oeuvre, nous détaillons chacun d'entre eux dans la liste ci-dessous :

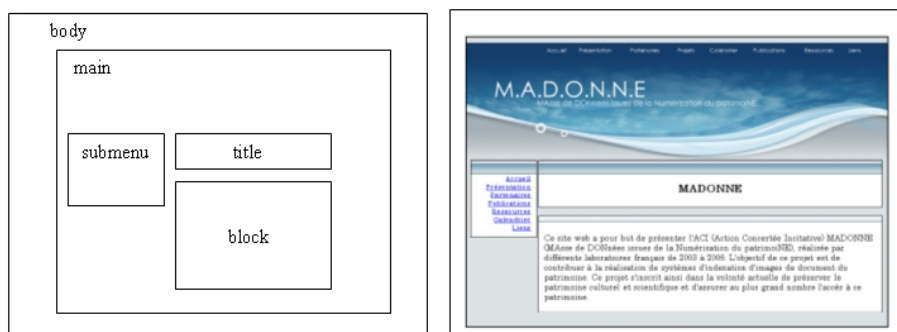


FIG. 2 – Composants principaux de la feuille de style *MADONNE*⁴

- block** : Le composant *block* définit une zone d'écriture. Il est de largeur pré-définie (610 px) et de hauteur variable. Il est par contre défini en position relative par rapport aux éléments précédents. Les différents composants *block* se positionnent donc successivement dans la page de façon automatique.
- title** : Le composant *title* est un *block* particulier présentant une bordure d'avantage marquée. Il sert à mentionner le titre de la page.
- submenu** : Le composant *submenu* exploite la même bordure que le composant *title*. Il est également de largeur pré-définie (100 px) et de hauteur variable. Il est par contre défini en position absolue afin d'être toujours aligné sur le composant *title* de la page.
- main** : Le composant de *main* est la racine de la page. L'utilisation de ce composant permet de délimiter la zone d'affichage de la page sous le navigateur.

Parallèlement à la définition de ces quatre composants principaux nous avons redéfini différents éléments de base du langage *HTML* : les headers `< h1 >` et `< h2 >`, les listes `< ul >` et les images `< img >`. La Figure 3 suivante illustre le rendu graphique de ces composants.

⁴Les visualisations présentées dans ce rapport ont été obtenues sous *Mozilla Firefox*.

Résultats

Les résultats du Consortium MADONNE sont de différents types :

- **Stratégiques** : réseau national de coordination de recherche
- **Scientifiques** : publications scientifiques de haut niveau
- **Opérationnels** : prototypes logiciels des laboratoires partenaires

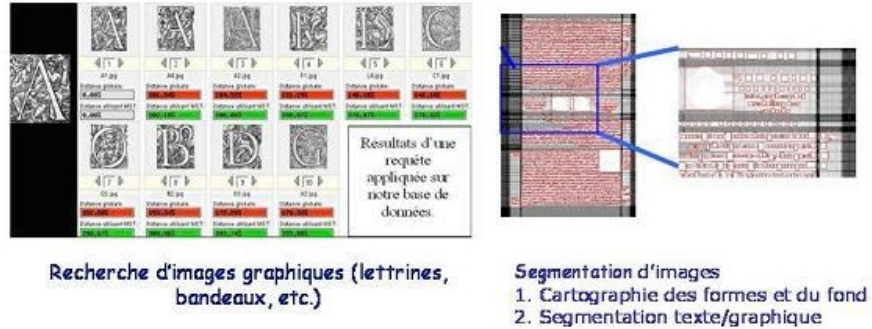


FIG. 3 – Composants *HTML* redéfinis dans la feuille de style *MADONNE*

2.4 Re-déclaration *Flash*

Le site initial comportait des animations *Flash* [4] qui ont été conservées au cours de la refonte du site. Cependant, nous avons re-déclaré ces animations à la manière dite *Flash Satay* [10]. En effet la déclaration initiale dite *twice-cooked*⁵ pêche par son manque de concision, la Figure 4 en donne un exemple. Elle est de plus non valide et ne peut être utilisée au sein d'un code *XHTML*⁶.

```
<object
classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com
/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
width="400" height="300" id="movie" align="">
<param name="movie" value="movie.swf">
<embed src="movie.swf" quality="high" width="400"
height="300" name="movie" align=""
type="application/x-shockwave-flash"
plug inspage="http://www.macromedia.com/go/getflashplayer">
</object>
```

FIG. 4 – Déclaration *twice-cooked*

⁵Littéralement "deux fois cuisinés" car combinant les déclarations *Netscape* et *Flash*.

⁶Nous reportons le lecteur à [11] sur ces aspects.

Une bonne compréhension des balises *Flash* permet de nettoyer cette déclaration *twice-cooked*. Ce nettoyage se résume en trois étapes principales⁷ listées ci-dessous. La Figure 5 suivante donne la déclaration nettoyée de la Figure 4.

1. La suppression de la balise *embed* qui est propre au navigateur *Netscape*.
2. La ré-écriture de l'attribut *classid* de la balise *object* en attribut *type MIME*⁸ supporté par le lecteur *Flash Macromedia*⁹.
3. La suppression de l'attribut *codebase* de la balise *object* utilisé de façon incorrecte dans la déclaration *twice-cooked*.

```
<object
type="application/x-shockwave-flash" data="movie.swf"
width="400" height="300">
<param name="movie" value="movie.swf" />
</object>
```

FIG. 5 – Déclaration *twice-cooked* nettoyée

Cette déclaration, quoi que valide sous *Netscape* et *Mozilla Firefox*, pose des problèmes sous *Internet Explorer*. En effet *Internet Explorer* ne procède à aucun streaming¹⁰, il attend alors que l'ensemble du clip soit chargé avant de procéder à sa lecture. Ceci permet une lecture correcte des clips de faibles tailles via des connexions hauts débits. Dans le cas contraire, la lecture du clip est alors différée ce qui rend cette déclaration peu applicable en pratique.

La solution pour résoudre ce problème consiste alors à créer un autre clip de faible taille (ou clip conteneur) qui, à partir d'une première fenêtre, charge le clip initial. La Figure 6 suivante donne un exemple d'appel d'un clip container nommé "*c.swf*". [10] précise alors : "*Peu importe que votre clip soit fait de boeuf, de poulet, ou de porc, vous aurez toujours besoin de l'embrocher et de le plonger dans la sauce pour que cela marche. Nous appelons cela la méthode Satay*"^{11,12}.

⁷Nous reportons le lecteur à [10] pour plus de détails sur ces étapes.

⁸Multipurpose Internet Mail Extensions

⁹<http://www.adobe.com/>

¹⁰Envoi de donnée au fur à mesure de la lecture.

¹¹"No matter if your movie is made from beef, chicken, or pork, you still need to skewer it and dip it in the sauce to make it work. We call this the Satay Method".

¹²Sauce d'origine thaïlandaise qui accompagne les brochettes de viande.


```

<object type="application/x-shockwave-flash"
data="c.swf?path=movie.swf"
width="400" height="300">
<param name="movie"
value="c.swf?path=movie.swf" />
</object>

```

FIG. 6 – Déclaration *Flash Satay*

2.5 Exemple d'utilisation

Dans les sections précédentes nous avons présenté la refonte *CSS* et la re-déclaration des animations *Flash* du site web *MADONNE*. Dans cette section nous donnons un court exemple d'utilisation des différents composants web présentés précédemment. Le code ci-dessous donne un exemple de page mettant en oeuvre d'une applet *Java* [12]. La Figure 7 donne la visualisation résultante.

```

<html>

<head>
  <title>Exemple</title>
  <link rel="stylesheet" type="text/css" href="sky.css"></link>
</head>

<body><div class="main">

<object type="application/x-shockwave-flash"
data="c.swf?path=sky/menu.swf" width="772" height="215">
  <param name="movie" value="sky/menu.swf"></param>
</object>

<div class="title">
  <h1>Exemple</h1>
</div>

<div class="submenu">
  <a href="exemple.html#introduction">Introduction</a>
</div>

<div class="block">
  <a name="introduction"></a>
  <h2>Introduction</h2>
  <p>
    Cette page présente un exemple de mise en oeuvre de la feuille de style

```

```

        utilisée pour la conception du site web Madonne.
        Voici un exemple d'appel d'Applet.
    </p>
    <p align="middle">
        <applet code="Exemple.class" width=300 height=60></applet>
    </p>
</div>

</div></body>

</html>

```



FIG. 7 – Exemple de mise en oeuvre

3 Gestion du contenu

3.1 Introduction

Dans cette section nous présentons la gestion du contenu effectuée sur le site web *MADONNE*. Celle-ci concerne deux aspects principaux : conception de pages *HTML* et base de données bibliographiques. Nous aborderons chacun de ces deux aspects dans les sous-sections suivantes (3.2) et (3.3).

3.2 Conception de pages *HTML*

Relativement aux aspects **bilan d'usage**, **ajout de contenu** et **bilan d'activité** présentés page 3 nous avons entrepris la conception et l'ajout de différentes pages *HTML* au site web *MADONNE*. La liste suivante en donne le détail.

Page d'accueil : Résume le projet et pointe sur les documents principaux.

Page de présentation : Présentation du projet.

Page de sous-projets : Description des sous-projets.

Page de calendrier : Dresse le calendrier des événements du projet et pointe sur les documents associés.

Page de ressources : Décrit et regroupe différentes ressources mises à disposition par le Consortium (logiciels, bases de données, etc.).

Page de liens : Présente différents sites web en rapport avec le projet.

3.3 Base de données bibliographiques

En complément des pages *HTML* présentées précédemment nous avons également conçu une page de publications. Cette dernière référence les différentes publications (article, revue, mémoire, thèse, etc.) des membres du Consortium et fournit les fichiers électroniques associés (*pdf*, *doc*, etc.). De notre point de vue il existe trois solutions pour la conception de ce type de page.

1. Simple édition d'une page *HTML*.
2. Constitution d'une base de données bibliographiques puis génération de la page *HTML* associée.
3. Édition collaborative en ligne.

Pour notre part nous avons retenu la solution 2. En effet la solution 1 semble à exclure dès lors que le nombre de publication devient trop important, des erreurs de déclaration peuvent alors se produire. Concernant la solution 3 celle-ci est souvent mise en oeuvre dans les sites web de laboratoires^{13,14}. Cependant, elle dispose de certains inconvénients pour nos besoins que nous listons ci-dessous.

- L'édition collaborative utilise des technologies dynamiques *PHP/MySQL* non retenues pour la refonte du site web *MADONNE*.
- En pratique les auteurs déclarent difficilement leurs publications de leur propre chef en dehors de leurs pages personnelles, il est donc souvent nécessaire de les "collecter".
- Les auteurs utilisent des nomenclatures différentes qu'il est nécessaire d'homogénéiser.

¹³<http://psiserver.insa-rouen.fr/psi/>

¹⁴<https://liris.cnrs.fr/>

- Certains auteurs référencent mal leurs publications, il nécessaire de vérifier et corriger leurs déclarations.
- Les auteurs déclarent souvent par "excès", il est nécessaire de filtrer certaines publications annexes.

Concernant la solution 2 celle-ci repose sur l'utilisation de logiciels de gestion de base de données bibliographiques. Ces logiciels proposent différentes fonctionnalités : édition de la base, tri automatique, recherche sur mot-clé, plugin éditeur de texte, gestion de la mise en forme, etc. Ils permettent ainsi une gestion aisée de la base de données bibliographiques. De notre point de vue on peut grouper ces logiciels en deux catégories.

1. Les premiers sont destinés aux professionnels de la documentation. Ils sont pour la plus part commerciaux, appliqués à de grandes bases et utilisent des formats de données propriétaires. Nous en citons ici quelques-uns¹⁵ : *Biblioscape*, *Bookends Plus*, *Library Master*, etc.
2. Les seconds sont destinés à la communauté informatique et recherche. Ils sont basés sur le format de données ouvert *BibTeX* et sont pour la plus part gratuits. Nous en citons ici quelques-uns¹⁶ : *Bib-it*, *BibTeX Manager*, *Jabref*, etc.

À la vue de nos besoins nous nous sommes orientés sur des logiciels de la seconde catégorie, ceux exploitant le format *BibTeX*. *BibTeX* est un format de fichier conçu pour le système de préparation de documents *LaTeX* [13]. Il permet l'incorporation aisée de citations au sein d'un document. Pour ce faire, *BibTeX* sépare les données bibliographiques de leurs informations de mise en forme. Ce mécanisme est en fait très similaire à l'utilisation conjointe *HTML/CSS*. La Figure 8 ci-dessous donne un exemple de code *BibTeX*. C'est un code de type textuel aisément interprétable. Ceci fait qu'il est facilement intégré au sein des logiciels. Les informations bibliographiques décrites dans ce fichier respectent un ensemble de règles syntaxiques. Chaque entrée décrit un type de publication (article, thèse, ...) et se compose de plusieurs champs (auteur, titre, journal, éditeur, etc.).

```
@BOOK{Lampport1986,
  Author      = {L. Lamport},
  Publisher   = {Addison-Wesley},
  Title       = {LaTeX: A Document Preparation System},
  Year        = {1986},
  ISBN        = {0-201-15790-X}}
```

FIG. 8 – Exemple de code *BibTeX*

¹⁵<http://users.ox.ac.uk/~ctitext2/service/workshop/bib-overview.html>

¹⁶<http://en.wikipedia.org/wiki/BibTeX>

Il existe un grand nombre¹⁶ de logiciels permettant la gestion de bases de données bibliographiques au format *BibTeX*. Dans le cadre de notre projet¹⁷ nous en avons retenu deux : *BibTeX Manager*¹⁸ pour l'édition de la base de données bibliographiques et *BibTeX2HTML*¹⁹ pour la génération de la page *HTML* associée.

4 Re-définition de la charte graphique

4.1 Introduction

Dans une dernière étape nous avons entrepris des travaux re-définition de la charte graphique du site web *MADONNE*. En effet, la conception d'une charte graphique nécessite des compétences d'ergonome et de graphiste [14]. La charte graphique existante a été conçue de façon arbitraire par des informaticiens [1]. Suite à un sondage réalisé²⁰ auprès des membres du Consortium *MADONNE* elle a été jugée comme peu adéquate avec la thématique du patrimoine. La Figure 9 suivante illustre ce manque d'adéquation en comparant la page d'accueil du site web *MADONNE* avec celle de *Gallica*²¹.



FIG. 9 – Comparaison de chartes graphiques

Nous déclarant également non compétent sur ces aspects nous avons proposé courant Novembre 2005 un sujet en Licence Professionnelle Création Multimédia²² de l'Université de La Rochelle. Malheureusement ce dernier n'a pas été pourvu, nous nous sommes donc résolu à conserver la charte graphique existante en la reproduisant durant la refonte *CSS* du code. Cependant, de façon à amorcer des travaux de re-définition de la charte graphique nous avons doté le site d'un mécanisme de styles alternatifs. Nous développons chacun de ces deux aspects dans les sous-sections suivantes (4.2) et (4.3).

¹⁷Une comparaison exhaustive des solutions logicielles sortirait du cadre de ce rapport.

¹⁸<http://www.imamod.ru/vab/bibtexmng/>

¹⁹<http://www.lri.fr/filiatr/bibtex2html/>

²⁰Par entretien téléphonique de Octobre à Novembre 2005.

²¹<http://gallica.bnf.fr/>

²²<http://www.univ-lr.fr/formations/pro/lpcm/>

4.2 Reproduction CSS

Tout d'abord, nous avons conservé la charte graphique existante en la reproduisant durant la refonte *CSS* du code (voir page 4). En effet, le site web *MADONNE* est en ligne depuis début 2004 [1]. Sa charte graphique constituait donc un repère pour ses utilisateurs et en l'absence d'étude d'ergonomie et graphique il semblait opportun de la conserver. Cependant, celle-ci a été reproduite dans le respect de la norme *CSS* et dans les limites de ses possibilités [8]. En effet, le site initial proposait des bords courbés pour encadrer les zones de texte comme l'illustre la Figure 10 (a) suivante. Ces bords ont été constitués par utilisation abusive de l'élément *table* en *HTML* et l'insertion d'images [5]. Ils étaient alors sujets à des déformations lors de zoom sur les navigateurs comme le montre la Figure 10 (b). La création de bords courbés étant encore en cours de définition dans la norme *CSS level 3*²³ nous avons donc substitué ces derniers par des bords droits (c) durant la refonte.

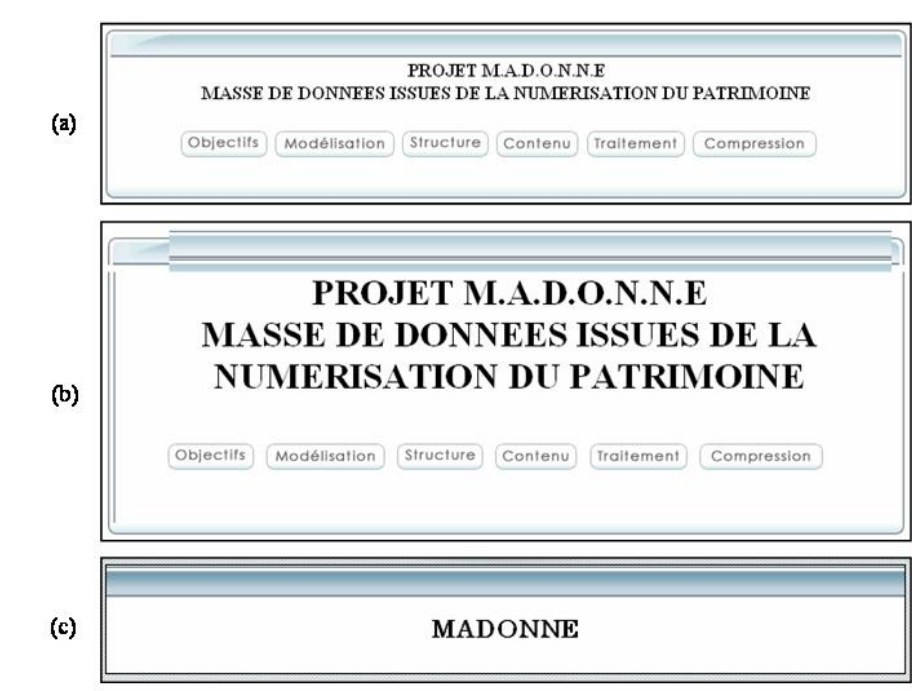


FIG. 10 – Divergence dans la reproduction de la charte graphique
(a) site initial (b) problème du zoom (c) site actuel

²³<http://www.w3.org/Style/CSS/current-work>

4.3 Styles alternatifs

Dans une deuxième étape, de façon à amorcer les travaux de re-définition de la charte graphique, nous avons doté le site d'un mécanisme de styles alternatifs. Le principe des styles alternatifs est de permettre à un visiteur d'afficher des mêmes contenus web de différentes manières [15]. De cette façon, il peut personnaliser le site selon ses besoins et préférences. Dans le cadre de notre travail, la perspective recherchée est de recueillir un sondage d'opinion sur les différents styles proposés afin de définir une charte graphique communément admise. Pour ce faire, nous avons défini les trois styles décrits dans la liste ci-dessous. Les Figures 11, 12 et 13 donne des exemples de rendu graphique de ces styles.

Classique : Le style HTML par défaut.

Ciel : Le style proposé dans le site initial.

Lettrine : Un style basée sur l'utilisation d'image de lettrine.

Il existe différentes solutions pour la mise en place des styles alternatifs [15]. Pour notre part nous avons retenu la solution proposée par [16]. Cette dernière est la plus communément utilisée, elle repose sur l'utilisation d'un switcher de styles *JavaScript*. Son implémentation se déroule en deux étapes décrites dans la liste ci-dessous. Nous présentons chacune d'entre elles par la suite.

1. Déclaration préférée et alternative des feuilles de styles.
2. Utilisation d'un d'un switcher de styles *JavaScript*.

Dans une première étape nous déclarons les différentes feuilles de style à utiliser en mode préféré et alternatif. En effet, une feuille de style en utilisation seule se déclare par défaut en mode permanent. La site suivante détaille les différences entre ces trois modes.

- Une feuille de style permanente (persistent style sheet) s'appliquera dans tous les cas au document quelques-soient les éventuelles autres feuilles de styles appliquées.
- Une feuille de style préférée (preferred style sheet) s'appliquera au document si aucune feuille de style alternative n'a été sélectionnée.
- Une feuille de style alternative (alternate style sheet) peut être sélectionnée par l'utilisateur pour remplacer une feuille de style préférée ou alternative.

La Figure 14 ci-dessous donne un exemple de déclaration d'une feuille de style en mode préféré (*Classic*) et deux autres en mode alternatif (*sky* et *dropcap*). La distinction de déclaration entre les deux réside dans la valeur de l'attribut *rel*.

MADONNE

Ce site web a pour but de présenter le projet MADONNE (MAsse de DONnées issues du Patrimoine), réalisé par différents laboratoires français de 2003 à 2006. L'objectif de ce projet est de contribuer à la réalisation de systèmes d'indexation d'images de document du patrimoine. Ce projet s'inscrit ainsi dans la volonté actuelle de préserver le patrimoine culturel et scientifique et d'assurer au plus grand nombre l'accès à ce patrimoine.

Quelques documents de synthèse:

- [Rapport mi-parcours](#)
- [Poster MADONNE](#)

FIG. 11 – Style *Classique*



FIG. 12 – Style *Ciel*



FIG. 13 – Style *Lettrine*


```

<link rel="stylesheet" type="text/css"
href="classic.css" title="classic"></link>
<link rel="alternative stylesheet" type="text/css"
href="sky.css" title="sky"></link>
<link rel="alternative stylesheet" type="text/css"
href="dropcap.css" title="dropcap"></link>

```

FIG. 14 – Déclaration en mode préféré/alternatif de feuilles de style

La déclaration en mode alternatif des styles n'est pas suffisante pour effectuer l'implémentation. En effet, différents problèmes persistent en majeure partie liés aux défaillances des navigateurs web²⁴ :

- Les styles alternatifs ne sont pas supportés par *Internet Explorer*.
- Les styles alternatifs sont appliqués de manière non permanente par les navigateurs *Mozilla Firefox* et *Netscape*.
- Les styles alternatifs ne prennent pas en compte les animations *Flash*.

La solution pour résoudre ces différents problèmes consiste à recourir à différentes fonctions *JavaScript*. Nous les présentons²⁵ ci-dessous.

setActiveStyleSheet : Cette fonction permet d'appliquer un style sélectionné à la page.

```

function setActiveStyleSheet(title) {
  var i, a, main;
  for(i=0; (a = document.getElementsByTagName("link")[i]); i++) {
    if(a.getAttribute("rel").indexOf("style") != -1 && a.getAttribute("title")) {
      a.disabled = true;
      if(a.getAttribute("title") == title) a.disabled = false;
    }
  }
}

```

getActiveStyleSheet, **createCookie**, **onunload** : Ces fonctions permettent de stocker le style en cours afin de le rendre persistant d'une page à l'autre sur le site. Pour ce faire un cookie est utilisé afin de le stocker. La méthode **getActiveStyleSheet** permet alors récupérer le style en cours appliqué à la page. Les fonctions **createCookie**, **onunload** permettent elles de créer le cookie d'y stocker le style.

²⁴Nous nous limitons ici aux navigateurs *Internet Explorer*, *Mozilla Firefox* et *Netscape*.

²⁵Nous les présentons sommairement ici et reportons le lecteur à [16] pour plus de détails.

```

function getActiveStyleSheet() {
    var i, a;
    for(i=0; (a = document.getElementsByTagName("link")[i]); i++) {
        if(a.getAttribute("rel").indexOf("style") != -1 && a.getAttribute("title")
            && !a.disabled) return a.getAttribute("title");
    }
    return null;
}
function createCookie(name,value,days) {
    if (days) {
        var date = new Date();
        date.setTime(date.getTime()+(days*24*60*60*1000));
        var expires = "; expires="+date.toGMTString();
    }
    else expires = "";
    document.cookie = name+"="+value+expires+"; path=/";
}
window.onunload = function(e) {
    var title = getActiveStyleSheet();
    createCookie("style", title, 365);
}

```

readCookie, main : Ces fonctions permettent de lire le cookie et de sélectionner le style stocké.

```

function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for(var i=0;i < ca.length;i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
    }
    return null;
}
/* main */
var cookie = readCookie("style");
setActiveStyleSheet(title);

```

getPreferredStyleSheet, onload : *Internet Explorer* ne supporte pas les styles alternatifs. Il est donc nécessaire de "forcer" le chargement du style préféré à l'ouverture de la page. Pour cela on utilise la fonction **getPreferredStyleSheet** (très proche de la fonction **getActiveStyleSheet**) en combinaison avec la fonction **setActiveStyleSheet**. On utilise alors ces deux fonctions durant le chargement de la page **onload** pour forcer la sélection du style préféré.

```
function getPreferredStyleSheet() {
    var i, a;
    for(i=0; (a = document.getElementsByTagName("link")[i]); i++) {
        if(a.getAttribute("rel").indexOf("style") != -1
            && a.getAttribute("rel").indexOf("alt") == -1
            && a.getAttribute("title")
        ) return a.getAttribute("title");
    }
    return null;
}
window.onload = function(e) {
    var cookie = readCookie("style");
    var title = cookie ? cookie : getPreferredStyleSheet();
    setActiveStyleSheet(title);
}
```

insertSampleMovie : Les styles alternatifs ne permettent pas le chargement dynamique d'animations *Flash* en fonction du style sélectionné. Pour ce faire nous avons développé une fonction complémentaire **insertSampleMovie**. Cette dernière repose sur l'utilisation des fonctions **readCookie** et **getPreferredStyleSheet**. Son utilisation implique la déclaration d'un composant *CSS* spécifiant les dimensions de l'animation. En effet, chacune des animations présente des dimensions différentes. Un chargement dynamique implique donc une externalisation de ces dimensions au sein des feuilles de style. Le code ci dessous donne un exemple de déclaration d'un composant *menu*.

```
function insertSampleMovie(movie) {
    var cookie = readCookie("style");
    var title = cookie ? cookie : getPreferredStyleSheet();
    var string;
    string = '<object type="application/x-shockwave-flash"
class="' + movie + '" data="templates/c.swf?path=templates/'
+ title + '/' + movie + '.swf">';
    document.write(string);
    string = '<param name="movie" value="templates/' + title
+ '/' + movie + '.swf"/>';
    document.write(string);
    document.write('</object>');
}
```

```
object.menu { position:relative; width:772px; height:215px; }
```

Dans une dernière étape il est nécessaire de faire fonctionner ces différentes fonctions ensemble. Il s'agit pour cela tout d'abord de charger le script *JavaScript* en entête de la page *HTML*.

```
<script language="JavaScript" src="switchcss.js"></script>
```

Ensuite, un menu cliquable doit être proposé à l'utilisateur afin d'y sélectionner les styles. Le code ci dessous donne un exemple de lien à mettre en oeuvre.

```
<a onclick="setActiveStyleSheet('classic');  
return false;" href="#">Classique</a>
```

5 Conclusion

Dans ce rapport nous avons présenté nos travaux de refonte du site web *MADONNE*. Ces travaux concernent différents aspects : suppression de la partie dynamique *PHP/MySQL*, refonte *CSS* du code, re-déclaration des animations *Flash*, conception de pages *HTML*, mise en place d'une base de données bibliographiques et implémentation d'un mécanisme de styles alternatifs. Au travers ces travaux le site a évolué selon deux axes principaux. Premièrement en terme de conformité du code. En effet, via la refonte *CSS* et de la re-déclaration des animations *Flash* le code a été rendu conforme aux normes en vigueur et de plus optimisé. De plus, il a été sécurisé de part la suppression de la partie dynamique *PHP/MySQL*. Le site a évolué ensuite en terme de contenu via la conception et l'ajout de pages *HTML* et la mise en place d'une base de données bibliographiques. De cette manière, le site répond aujourd'hui au positionnement actuel du projet *MADONNE*.

Les perspectives d'évolution de ce site sont doubles. Premièrement comme nous l'avons évoqué dans ce rapport la charte graphique actuellement mise en place est peu adéquate avec la thématique du patrimoine. Elle doit donc être redéfinie au travers d'études d'ergonomie et de graphisme [14]. Nous avons déjà entrepris des travaux dans ce sens au travers de la refonte *CSS* du site et de la mise en place d'un mécanisme de styles alternatifs. Dans un deuxième temps aucun travail de référencement n'a été effectué pour ce site (inscription aux annuaires web, déclaration de méta-données, optimisation de liens, etc). Un bon point d'entrée il nous semble pour ce travail peut être trouvé dans [17].

6 Annexes

6.1 Feuille de style *CSS*

```
/******  
  
Sky stylesheet for the MADONNE website  
  
@author Mathieu Delalandre @date   September 2006  
  
*****/  
  
/* root element */  
  
body { background-color:#FFFFFF; font-family:century;  
word-spacing:1pt; text-align: left; font-size:11pt; }  
  
div.main { position:absolute; width:772px;  
margin-left:15%;  
margin-right:15%;  
padding-top:10px; padding-bottom:10px;  
background-image:url("sky/background.jpg"); border: 2px solid  
#000000; }  
  
/* submenu */  
  
div.submenu {  
  
position:absolute; width:100px; top:225px;  
left:1%;  
  
padding-top:25px; padding-bottom:10px; padding-left:10px;  
padding-right:10px;  
  
background-repeat: repeat-x; background-image:url("sky/title.jpg");  
border: 1px ridge #000000; background-color:#FFFFFF;  
  
font-size:8pt; word-spacing:2pt; letter-spacing:1pt; text-align:  
right; }  
  
/* title */  
  
div.title{ position:relative; width:610px;  
margin-left:17%;  
padding-top:25px; padding-bottom:10px; padding-left:10px;  
padding-right:10px;
```

```

background-repeat: repeat-x; background-image:url("sky/title.jpg");
border: 1px ridge #000000; background-color:#FFFFFF; }

/* block */

div.block { position:relative; width:610px;
margin-left:17%;
margin-top:2%;
margin-bottom:2%;
padding-top:15px; padding-bottom:5px; padding-left:10px;
padding-right:10px;

background-repeat: repeat-x; background-image:url("sky/block.jpg");
border: 1px ridge #000000; background-color:#FFFFFF;

}

/* html elements */

h1{ font-size:15pt; font-weight:bold; text-align:center; }

h2{ font-size:14pt; font-weight:bold; }

ul { list-style-image:url("sky/list.jpg"); }

img { margin-left:10px; margin-right:10px; }

```

6.2 Makefile *BibTeX2HTML*

```

cd .. publications\bibtex2html -o publications\body -a -nodoc
-nobibsource publications.bib copy
publications\up.html+publications\body.html+publications\bottom.html
publications.html

```

6.3 Switcher de styles *JavaScript*

```

function setActiveStyleSheet(title) {
    var i, a, main;
    for(i=0; (a = document.getElementsByTagName("link")[i]); i++) {
        if(a.getAttribute("rel").indexOf("style") != -1 && a.getAttribute("title")) {
            a.disabled = true;
            if(a.getAttribute("title") == title) a.disabled = false;
        }
    }
}

```

```

function getActiveStyleSheet() {
    var i, a;
    for(i=0; (a = document.getElementsByTagName("link")[i]); i++) {
        if(a.getAttribute("rel").indexOf("style") != -1 && a.getAttribute("title")
            && !a.disabled) return a.getAttribute("title");
    }
    return null;
}

function getPreferredStyleSheet() {
    var i, a;
    for(i=0; (a = document.getElementsByTagName("link")[i]); i++) {
        if(a.getAttribute("rel").indexOf("style") != -1
            && a.getAttribute("rel").indexOf("alt") == -1
            && a.getAttribute("title")
            ) return a.getAttribute("title");
    }
    return null;
}

function createCookie(name,value,days) {
    if (days) {
        var date = new Date();
        date.setTime(date.getTime()+(days*24*60*60*1000));
        var expires = "; expires="+date.toGMTString();
    }
    else expires = "";
    document.cookie = name+"="+value+expires+"; path=/";
}

function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for(var i=0;i < ca.length;i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
    }
    return null;
}

window.onload = function(e) {
    var cookie = readCookie("style");
    var title = cookie ? cookie : getPreferredStyleSheet();
    setActiveStyleSheet(title);
}

```

```

}

window.onunload = function(e) {
    var title = getActiveStyleSheet();
    createCookie("style", title, 365);
}

function insertSampleMovie(movie) {
    var cookie = readCookie("style");
    var title = cookie ? cookie : getPreferredStyleSheet();
    var string;
    string = '<object type="application/x-shockwave-flash"
class="' + movie + '" data="templates/c.swf?path=templates/'
+ title + '/' + movie + '.swf">';
    document.write(string);
    string = '<param name="movie" value="templates/' + title
+ '/' + movie + '.swf"/>';
    document.write(string);
    document.write('</object>');
}

var cookie = readCookie("style"); var title = cookie ? cookie :
getPreferredStyleSheet(); setActiveStyleSheet(title);

```

6.4 Mise en oeuvre des styles alternatifs

```

<html>

<head>
    <title>Site Madonne</title>
    <link rel="stylesheet" type="text/css"
href="templates/classic.css" title="classic"></link>
    <link rel="alternative stylesheet" type="text/css"
href="templates/sky.css" title="sky"></link>
    <link rel="alternative stylesheet" type="text/css"
href="templates/dropcap.css" title="dropcap"></link>
    <script language="JavaScript" src="switchcss.js"></script>
</head>

<body><div class="main">

<script language="JavaScript">
    insertSampleMovie("menu");
</script>

<div class="submenu">

```



```

    <a onclick="setActiveStyleSheet('classic'); window.location.reload();
    return false;" href="#">Classique</a>
    <a onclick="setActiveStyleSheet('sky'); window.location.reload();
    return false;" href="#">Ciel</a>
    <a onclick="setActiveStyleSheet('dropcap'); window.location.reload();
    return false;" href="#">Lettrine</a>
</div>

<div class="title">
    <h1>MADONNE</h1>
</div>

<div class="block">
    <p>
        <span class="dropC"><span>C</span></span>e site web a pour but
        de présenter le projet MADONNE
        réalisé par différents laboratoires français de 2003 à 2006.
        L'objectif de ce projet est de contribuer à la réalisation
        de systèmes d'indexation d'images de document du patrimoine.
        Ce projet s'inscrit ainsi dans la volonté actuelle de préserver
        le patrimoine culturel et scientifique et d'assurer
        au plus grand nombre l'accès à ce patrimoine.
    </p>
</div>

</div></body>

</html>

```

Références

- [1] P. Dezanneau, P. Dujon, P. Ferret, D. Fisson, L. Paviot, A. Verneuil, Réalisation d'un site web pour le projet aci madonne, Tech. rep., Laboratoire L3i, Université la Rochelle, France (2004).
- [2] O. B. University, HTML Primer, Oxford Brookes University, 2003.
- [3] L. Lancker, PHP - MySQL (2003).
- [4] Macromedia, Utilisation de flash, Tech. Rep. ZFL70M200F, Macromedia, Inc (2004).
- [5] D. Shafer, HTML Utopia : Designing Without Tables Using CSS, 1st Edition, SitePoint Pty Ltd, 2003.
- [6] W3C, Cascading style sheets, level 1, Tech. Rep. REC-CSS1-961217, World Wide Web Consortium (W3C) (1996).
- [7] W3C, Html 4.01 specification, Tech. Rep. PR-html40-1999082, World Wide Web Consortium (W3C) (1999).
- [8] W3C, Cascading style sheets, level 2, Tech. Rep. REC-CSS2-19980512, World Wide Web Consortium (W3C) (2001).
- [9] O. B. University, CSS Primer, Oxford Brookes University, 2003.
- [10] D. Mclellan, Flash satay : Embedding flash while supporting standards, A List Apart, N° 154 (2002).
- [11] W3C, Xhtml 1.0 the extensible hypertext markup language (second edition), Tech. Rep. REC-xhtml1-20020801, World Wide Web Consortium (W3C) (2000).
- [12] Sun, Programmer's guide to the java 2d api, Tech. rep., Sun Microsystems, Inc (2001).
- [13] L. Lamport, LaTeX : A Document Preparation System, Addison-Wesley, 1986.
- [14] F. Millerand, O. Martial, Guide pratique de conception et d'évaluation ergonomique de sites web, Tech. rep., Centre de Recherche Informatique de Montréal (CRIM), Montréal, Canada (2001).
- [15] L. Denis, Feuilles de styles permanentes, alternatives et préférées en (x)html, Blog and Blues (2004).
- [16] P. Sowden, Alternative style : Working with alternate style sheets, A List Apart, N° 126 (2001).
- [17] S. Caron, Intérêt du référencement et des référenceurs pour un site internet d'entreprise, Master's thesis, Ecole Hec-Entrepreneurs, Paris, France (2003).

Table des figures

1	Exemple de code <i>HTML</i> vs <i>HTML/CSS</i>	5
2	Composants principaux de la feuille de style <i>MADONNE</i> ²⁶	6
3	Composants <i>HTML</i> redéfinis dans la feuille de style <i>MADONNE</i>	7
4	Déclaration <i>twice-cooked</i>	7
5	Déclaration <i>twice-cooked</i> nettoyée	8
6	Déclaration <i>Flash Satay</i>	9
7	Exemple de mise en oeuvre	10
8	Exemple de code <i>BibTeX</i>	12
9	Comparaison de chartes graphiques	13
10	Divergence dans la reproduction de la charte graphique	14
11	Style <i>Classique</i>	16
12	Style <i>Ciel</i>	16
13	Style <i>Lettrine</i>	16
14	Déclaration en mode préféré/alternatif de feuilles de style	17