|  |  |
| --- | --- |
|  | **Master Informatique IHM Promotion 2011-2012** |

Rapport de stage présenté   
par

Nicolas Reitz

En Vue de l’obtention du Master Informatique

Spécialité Interaction/Interface Homme Machine

|  |  |
| --- | --- |
| **Stage effectué A Atos** | 8 rue Lafayette  57000 Metz |

# Remerciement

Je remercie Atos de m’avoir accueilli durant mon stage de fin d’étude pour l’obtention du master informatique IHM.

Je remercie Bertrand Déchery et Aline Walter de m’avoir encadré, ainsi que Loïc Gangloff pour l’aide apporté tout au long du stage.

Je remercie ???, professeur à l’université de m’avoir été encadré.

# Sommaire

Remerciement 1

Sommaire 2

Table des illustrations 4

Introduction 7

Présentation d’Atos 8

Sujet 9

Cycle de vie d’un projet 10

Le Portail des Lorrains (PDL) 12

Présentation de Jahia 12

Développement 12

Syntaxe 13

Module 14

Template 14

Portlet 14

Les trois types de requêtes possibles sous Jahia 14

Travaux relatifs au Portail des Lorrains - version 1 15

FML 15

Version 2.0 du Portail des Lorrains 19

Gestion de projet - Déroulement du développement de la version 2 19

Outils utilisés 20

Problématique du découpage de certains éléments 20

Le patron du site 21

En-tête du site 21

Barre permanente 23

La cartographie des actions régionales 27

La newsletter 29

Recherche Synomia 31

Les actualités 32

Les Articles 32

Galerie Photo & Vidéo 33

Les Tests 33

En développement 33

En intégration 34

Référencement et Accessibilité 34

Référencement SEO 34

Accessibilité 36

Conclusion 37

Table des Annexes 38

Annexe A

Annexe i - Commentaire jour à jour A

Annexe ii - Maquette du portail des Lorrains v2 K

Annexe iii - Autre L

Table des matières A

# Table des illustrations

Figure 1 - Cycle de vie d'un projet 10

Figure 2 - Schéma explicatif se basant sur une vue éclipse 13

Figure 3 - Détail d'un module 13

Figure 4 - Visualisation de la mise en forme sous deux colonnes 15

Figure 5 - Bouton Facebook visible par la région lorraine 16

Figure 6 - Bouton Facebook visible par un internaute lambda 16

Figure 7 - Schématisation de la problématique 16

Figure 8 – Comportement classique de la fenêtre de commentaire après clic sur le bouton « J'aime » 17

Figure 9 - Illustration de la méthode image 20

Figure 10 - Slider sans focus 22

Figure 11 - Slider avec focus 22

Figure 12 - Etat initial 22

Figure 13 - Le bandeau de navigation t1 22

Figure 14 - Le bandeau de navigation t2 22

Figure 15 - Fonctionnement avec et sans JavaScript 23

Figure 16 - Maquette de la barre 23

Figure 17 - Information visible avec JavaScript 24

Figure 18 - Information sans JavaScript 24

Figure 19 - Information avec JavaScript 24

Figure 20 - Information sans JavaScript 24

Figure 21 - Entrée par cible 25

Figure 22 - Fenêtre de conception d'un service en ligne non navigable (lien non navigable) 25

Figure 23 - Ancien formulaire 26

Figure 24 - Nouveau formulaire 26

Figure 25 - Formulaire avec JavaScript désactivé 26

Figure 26 - Espace privé 26

Figure 27 - Slider sans focus 26

Figure 28 - Slider avec focus sur un item 26

Figure 29 - Slider sans focus 27

Figure 30 - Slider avec focus sur un item 27

Figure 31 – Publicité non déployée 27

Figure 32 - Publicité déployée 27

Figure 33 - Zone flottante en position par défaut 29

Figure 34 - Zone flottante dans une autre position 29

Figure 35 - Inscription à la newsletter 30

Figure 36 - Administration de la newsletter 30

Figure 37 - Choix des abonnements 30

Figure 38 - Formulaire d'accès rapide à la newsletter 30

Figure 39 - Communication formulaire / portlet 31

Figure 40 – Ancien formulaire résultat attendu 31

Figure 41 – Ancien formulaire résultat obtenu sur certains navigateurs 31

Figure 42 - Nouveau formulaire résultat uniforme sur les principaux navigateurs 31

Figure 43 – Aperçu du type « submit » 32

Figure 44 – Aperçu du type « image » 32

Figure 45 - Nouvelle vue de l’actualité à la une 32

Figure 46 – Ancienne vue des « n » dernières actualités 32

Figure 47 – Nouvelle vue des « n » dernières actualités 32

Figure 48 - Nouvelle vue des articles 33

Figure 49 – Ancienne actualité vidéo 33

Figure 50 – Nouvelle actualité vidéo 33

Figure 51 – Ancienne galerie 33

Figure 52 – Nouvelle galerie 33

# Introduction

Dans le cadre de l’obtention du diplôme de master informatique spécialité IHM, j’ai effectué un stage de 6 mois dans la SS2I Atos de l’agence de Metz.

Tout au long de ce rapport sont abordés certains points concernant la gestion de projet, ainsi que les moyens de communication utilisés chez Atos pour assurer un suivi, et garantir un service de qualité.

Ce rapport contient une courte partie concernant Jahia afin d’expliquer la logique de ce CMS et d’établir certaines notions présente dans la suite de ce document.

Par la suite, j’aborderai les différentes étapes de la vie du projet, l’analyse, le développement, les différents problèmes révélés par les tests, ainsi que certaines problématiques relatives à la spécialité IHM telles que l’accessibilité et les problèmes de conception. Pour finir le référencement sera abordé.

En Annexe i - Commentaire jour à jour figure le suivi des tâches effectuées au jour le jour.

# Présentation d’Atos

# Sujet

# Cycle de vie d’un projet

Atos utilise la méthode de développement agile et met en application le principe de développement sur SVN. Ci-dessous figure un schéma représentant la démarche de gestion de projet.

Si besoin

Commentaire

**Chez le client**

Cahier des charges / Spécifications

Développement local + fiche de tests

Développement sur la plateforme de développement (si besoin)

Test sur la plateforme d’intégration

Test en pré production

Mise en production

Test sur la plateforme de développement

Fiche Mantis

Livraison

Figure - Cycle de vie d'un projet

Le système de Mantis permet de faire le suivi de divers bogue, et nouvelles fonctionnalités souhaitées. Lorsqu’une fiche Mantis est publiée, les Spécifications fonctionnelles détaillées et Spécifications fonctionnelles générales sont mises à jour avec la demande qui est récapitulée soit dans un Dossier d’analyse fonctionnelle ou dans une spécification fonctionnelle détaillée propre à la fiche.

Les développements et les tests unitaires s’effectuent en local où est installée une version de Jahia en mode développement. Les plateformes de développement et d’intégration ont une version de production de Jahia. Ces plateformes se rapprochent des configurations du client et intègrent l’ensemble des modules ; les tests se déroulant sur ces environnements permettent de tester la compatibilité du développement avec les dernières versions des modules.

Les fiches de tests reprennent les fonctionnalités qui doivent être testées en adéquation avec les spécifications du cahier des charges. Dans un but d’optimisation seul est testé ce qui est impacté par le nouveau développement.

Une fois l’étape d’intégration effectuée, on établit un bon de commande, récapitulant différents éléments tels que les documents en relation, démarche d’installation, ce qu’il faut déployer, etc. puis on livre sous SVN les archives « war » générées lors de la compilation du développement.

# Le Portail des Lorrains (PDL)

Le portail des Lorrains est un site web demandé par la région lorraine, il a été développé avec le CMS Jahia par Atos dans le service TMA (Technologie et Maintenance Applicative) – XXNET. Ce service s’occupe de la maintenance des sites de la région lorraine et développe, en parallèle, les nouveaux projets demandés par la région.

## Présentation de Jahia

Une formation de deux jours en Jahia6.5 m'a été dispensée afin d'avoir des bases pour le développement du PDL. Le PDL est un site web administré par la région lorraine. Ce site utilise le CMS Jahia, basé sur le langage java. Ce CMS autorise l’ajout de vue et la manipulation de la base de données Jahia. En effet, il intègre une base de données de type JCR[[1]](#footnote-1) dont la spécificité est d’être orienté objet et est représenter sous forme d’arborescence. Les opérations les plus délicates restent l’élaboration d’éventuelles requêtes SQL afin de réaliser des vues particulières, en effet le JCR peut ne pas suffire dans certain cas. Le JCR est un système de stockage équivalent à une méta-base de données

Ce CMS contrôle la saisie de certaines informations afin qu’elles soient conformes à certaines normes d’accessibilité WCAG tel que l’attribut « alt » à renseigner pour les images et « title » pour les liens.

### Développement

Il est possible d’ajouter des éléments à Jahia. Ces éléments portent le nom de module, de template et de portlet. Le développement de ces composants se fait en Java, les vues sont gérées par des pages JSP. Jahia possède un mode édition et un mode « live ». Il est possible de faire un rendu « pauvre » pour l’édition et un contenu « riche » pour le mode « live » suivant le contexte d’utilisation du CMS. Le contenu de la page est à la discrétion du client qui construit ses pages avec l’aide l’éditeur mis à disposition par Jahia.

Nous utilisons Maven, afin de construire le fichier à déployer. C’est un outil pour la gestion et l'automatisation de production des projets Java.

Dans les parties suivantes, des explications sur les divers éléments abordés sont décrites.

### Syntaxe

On peut assimiler un module à un projet Maven. Un composant est une partie d’un module. Ce composant peut être caractérisé par différentes vues, pouvant posséder différents champs.



Les projets enfants

=

Modules

Fichier de préférence de déploiement

Fichier de configuration du projet parent

Figure - Schéma explicatif se basant sur une vue éclipse



Vue du composant

Fichier de configuration du module

Composants du module

Figure - Détail d'un module

### Module

Ajouter un module à Jahia, c’est ajouter une nouvelle famille de fonctionnalités. Un composant permet de réaliser le plus petit bloc de données identifiables. Par exemple, un article est un composant utilisant un titre, un contenu, des images ; un article votable est un composant nécessitant un titre, des descriptions courtes, éventuellement un lien et un bouton pour voter. Ces deux composants peuvent être réunis dans un même module (ici crlo – article)

### Template

Le template permet de définir un patron de vue pour une page, il définit donc les principales régions d’une page. Par exemple l’en-tête, le pied de page, la barre de menu pour une page classique. Ce patron est commun à toutes les pages du site. Le template est aussi la base pour la construction de nouvelle page.

### Portlet

Les portlets sont des applications autonomes qui permettent d’afficher du contenu dans une page web. Elles sont souvent émises par des servlets. Ces composants permettent de réaliser des services généralistes ou spécialisés tels que des annuaires, des moteurs de recherche, agenda, etc.

### Les trois types de requêtes possibles sous Jahia

#### Le SQL 2

Langage similaire à celui utilisé dans les bases de données. Les requêtes sont d’abord parsées et transformées en JQOM (Java Query Object Model) avant d’être exécutées. Du fait de son manque de maturité, il n’est pas optimisé pour les tâches complexes (jointure, etc.)

#### Le JQOM (Java Query Object Model)

Il s’agit d’une représentation en objet Java d’une requête. On peut construire ces requêtes à l’aide des bibliothèques de tags de recherche fournies par Jahia, ou bien directement à partir de code Java. Les requêtes SQL-2 et JQOM sont assez ressemblantes, sauf que les requêtes JQOM évitent l’étape de parsing et sont donc légèrement plus rapides.

#### Le XPATH

C’est le langage de requête le plus optimisé, mais il est aussi le plus compliqué à mettre en œuvre en raison de son écriture. L’écriture de ce type de requêtes ressemble au chemin d’une arborescence vers un fichier.

## Travaux relatifs au Portail des Lorrains - version 1

### FML

Le premier développement effectué consistait à créer un composant permettant aux visiteurs de voter pour un groupe de musique dans le cadre du festival des musiques lycéennes. Les contraintes données par le client étaient les suivantes :

L’affichage de « n » articles se fait sous forme de deux colonnes d’articles votables.



Composant article votable

Du module crlo - article

Figure - Visualisation de la mise en forme sous deux colonnes

Le vote via Facebook avec différent affichage selon que l’internaute est un utilisateur lambda ou la région lorraine

|  |  |
| --- | --- |
| Macintosh HD:Users:Nico:Desktop:Capture d’écran 2012-04-10 à 23.29.18.png | Macintosh HD:Users:Nico:Desktop:Capture d’écran 2012-04-10 à 23.29.18.png |
| Figure - Bouton Facebook visible par la région lorraine | Figure - Bouton Facebook visible par un internaute lambda |

Le module doit être générique pour être utilisé dans différents contextes.

Il découle de ceci et de la maquette réalisée par le client, les 5 éléments suivants. Un titre, une image, deux brèves descriptions ainsi qu’un lien vers une vidéo externe. Le titre est un champ obligatoire, en effet il discrimine l’article votable.

#### Conceptions — différentes solutions possibles :

##### Solution 1

Cette solution consiste à déléguer l’affichage (placement sur la page) au module même via l’utilisation de propriété CSS. Elle est simple à mettre en œuvre, en effet elle nécessite uniquement de développer un nouveau composant « article votable » et d’utiliser les propriétés CSS de sorte qu’un article occupe 50 pour cent de l’espace disponible. Ainsi le second composant peut venir se placer directement à côté. L’inconvénient de cette méthode est de détecter, dans le module courant, si celui-ci est le dernier élément. En effet, dans le cas d’un nombre impair d’éléments le prochain contenu est susceptible de se placer au mauvais endroit. De plus, il devient délicat de gérer ce problème si l’on souhaite faire des regroupements d’articles dans différentes catégories.

Cette solution, bien que bonne sur le principe, semble tout de même souffrir de quelques inconvénients qui empêchent la réutilisabilité dans différentes configurations.



Figure - Schématisation de la problématique

##### Solution 2

Cette solution consiste à déléguer l’affichage des articles à un module tiers. Cette méthode nécessite le développement d’un nouveau composant « article votable », mais aussi de développer une vue « 2colonnes » qui aurait pour rôle de structurer les articles en colonne et d’en afficher qu’un certain nombre. Cette solution permet de diminuer l’interaction avec les CSS et par la même occasion l’autoplacement d’un module comme expliqué ci-dessus. Cette méthode à l’avantage d’être itérative au contraire de la première solution, et donc de parfaitement maîtriser le positionnement du contenu.

##### Solution retenue

Afin de pouvoir contrôler correctement l’affichage, il a été choisi d’appliquer la deuxième solution.

#### Problèmes rencontrés

Le souci principal de ce développement a été le système de vote via Facebook. En effet, la demande du client détourne un peu l’utilisation du système de « J’aime ». La solution retenue pour masquer le compteur est de rajouter un « bandeau », de couleur identique au fond, au-dessus du compteur afin de le masquer. La seconde solution consistait à contraindre l’affichage au seul bouton. Cette méthode posait problème, car l’affichage de la zone de commentaire empiétait sur le bouton Facebook, ceci se produisant sur certains navigateurs (Internet Explorer et Firefox).



Figure – Comportement classique de la fenêtre de commentaire après clic sur le bouton « J'aime »

Le second souci est de détecter qui est connecté afin de déterminer si le compteur doit être affiché ou non. Dans le contexte d’utilisation du site, il n’y a pas de notion de compte utilisateur, on ne peut donc pas se baser sur l’identifiant de l’utilisateur courant. Néanmoins, Jahia utilise un système de gestion de rôles et de permissions dans l’administration de ses sites. Il suffit donc d’ajouter une permission spéciale à un rôle qui sera attribué à l’utilisateur chargé de comptabiliser les votes.

#### Les remarques du client

Le client a demandé l’ajout d’une fonctionnalité qui a échappé à notre attention. En effet, la demande d’afficher l’intégralité du texte masqué par les points de suspension « … » dans une bulle d’information a été faite. La solution de base consiste à utiliser l’attribut « title » néanmoins, cette solution pose problème, car le « title » est prévu pour se masquer au bout d’un certain temps. La balise « abbr » a donc été utilisée, elle présente les mêmes caractéristiques que l’attribut « title » à l’exception qu’elle ne masque pas le texte au bout d’un délai delta.

#### Évolution en parallèle au développement du Portail des Lorrains Version 2

En parallèle du développement de la version 2, une fiche Mantis a été ouverte afin d’effectuer une évolution mineure qui consiste à ajouter un lien permettant de télécharger la prestation du groupe de musique.

## Version 2.0 du Portail des Lorrains

La région lorraine a décidé de faire une mise à jour majeure de son site, publié en janvier 2012, afin de mieux répondre aux nouveaux besoins de la région. Cette mise à jour porte aussi bien sur le design et la charte graphique du site que sur les différents modules existants. De plus ce site étant commandé par une collectivité territoriale française, il doit respecter les normes d’accessibilités. La région demande le niveau AA du RGAA (Référentiel général d'accessibilité pour les administrations).

Une équipe de trois développeurs est prévue pour le développement de la version 2. Loïc Gangloff, Alexandre Lebeau ainsi que le rédacteur de ce rapport[[2]](#footnote-2).

Afin de simplifier toute manipulation JavaScript, le framework jQuery est utilisé, ainsi que de nombreux plug-ins jQuery. La région lorraine a fourni une charte de graphique, conçu par Prodaction afin de nous aider dans le développement des vues. Une des contraintes imposées par le client est que les fonctionnalités soient opérationnelles sous Internet Explorer[[3]](#footnote-3) et Firefox[[4]](#footnote-4).

### Gestion de projet - Déroulement du développement de la version 2

La priorité sur la TMA est donnée au développement de la version 2. En parallèle, le client affecte une priorité aux autres demandes afin que l’on puisse les traiter correctement.

Dans un premier temps, à partir d’un document du client (CCTP), nous avons fait du prototypage afin de vérifier la faisabilité en ce qui concerne les fonctionnalités. Une fois obtenu un résultat satisfaisant, les prototypages ont continué sous Jahia, afin de mesurer la complexité des changements et d’avoir une estimation du temps de travail nécessaire.

Lorsque le projet a pu réellement commencer, nous avons rédigé divers documents concernant la mise à jour du PDL. Ces documents servent à assurer le suivi des mises à jour du projet et nous assurent de notre compréhension vis-à-vis de la demande du client. Une étape importante dans l’accomplissement de ce projet, a été la communication avec le client afin de l’informer d’éventuelles incohérences (changement de charte graphique sur les titres, distinction entre les liens hypertextes et le contenu, etc.), d’obtenir des informations complémentaires par rapport à une fonctionnalité non décrite, ou demander des précisions lorsqu’un trop-plein d’information masquait la compréhension de la charte.

Dans un troisième temps, nous avons réalisé une maquette dynamique incluant les éléments de la charte graphique, afin que le client ait un aperçu du résultat, sur les jeux de couleurs (incluant la transparence), ainsi que la mise en page. La réalisation de cette maquette a permis de faciliter l’intégration dans Jahia en évitant de déployer les modules concernés par la modification de trop nombreuses fois. Enfin, nous avons intégré dans Jahia l’ensemble des demandes.

### Outils utilisés

Afin de pouvoir manipuler la charte graphique, au format PSD[[5]](#footnote-5), nous avons utilisé dans un premier temps le logiciel Gimp. Dans la constatation des différences de visualisation entre Gimp et Photoshop (notamment dans la nuance de couleur), nous sommes passés sous Photoshop : version d’évaluation.

Les développements, réalisés sous Éclipse, ainsi que les documents relatifs au projet sont stockés sous subversion sur les serveurs de la région lorraine.

### Problématique du découpage de certains éléments

La maquette à une résolution de 2000x3000, il a donc fallu faire des redimensionnements afin d’avoir un visuel esthétiquement correct pour des résolutions d’écran plus communes.

Dans la charte graphique, certains éléments, tels que des fonds pour menu déroulant, doivent avoir des bords arrondis en plus d’une transparence. Il existe différentes méthodes afin de faire ceci. Elles ne sont pas toutes compatibles avec les différents navigateurs existants.

Afin d’avoir un comportement correct sur le plus grand nombre de navigateurs, on peut utiliser diverses propriétés du CSS3 qui permettent d’afficher les arrondis ; ces propriétés sont malheureusement inefficaces sous Internet Explorer de même, le fond ne peut être remplacé par une couleur de fond de type RGBA[[6]](#footnote-6). Ces diverses observations impliquent l’utilisation d’une image dont le format autorise la transparence, par exemple le PNG.

Une autre contrainte implicite est que le fond soit adaptable à toute longueur de texte, pour être utilisable dans le plus grand nombre de contextes. Cette contrainte implique que l’on ne peut utiliser plusieurs composantes d’une image pour faire ce fond, en effet cette méthode nécessite des valeurs fixes pour avoir une imbrication correcte de toutes les composantes de l’image.

C:\Users\Nicolas\Desktop\Rapport Stage\image\Bouton de Base.png

Figure - Illustration de la méthode image

Une autre solution consiste à n’utiliser qu’un pixel et à le répéter sur tous les points d’abscisse et d’ordonnée de la zone. Cette solution permet de gérer correctement l’extension du fond. Le dernier problème est la gestion des arrondis. Deux solutions sont possibles. La première est de mettre une image ou un fond d'écran supplémentaire dans chaque coin de la zone ; le problème inhérent à cette méthode est que l’image de fond des coins soit identique à la couleur de fond environnant. Une autre solution consiste à créer un style pour Internet Explorer, on peut inclure ce nouveau style en utilisant la propriété CSS « behavior », en lui précisant un fichier HTC[[7]](#footnote-7). Cette méthode permet ainsi d’obtenir le même effet sous tous les navigateurs, dont Internet Explorer.

### Le patron du site

Nous avons profité du projet PDL v2 pour revoir le modèle de base de toutes les pages et poser les bases et normes de conception. Parmi les normes posées, quelques-unes portent sur les feuilles de styles CSS, par exemple les styles généraux sont insérés avant les styles des modules. Ceci permet dans de nombreuses circonstances d’éviter l’emploi de hack CSS. Les hacks CSS[[8]](#footnote-8), lorsqu’ils sont nécessaires, sont remplacés par une méthode plus propre : les CSS conditionnelles, qui permettent de définir des styles particuliers pour les différentes versions d’Internet Explorer tout en optimisant la lisibilité du code.

La conséquence d’utiliser plusieurs modules se servant du JavaScript est la duplication du code qui peut entraîner des complications imprévues, par exemple l’association de deux actions identiques sur un même bouton. Nous avons donc posé la convention d’employer une balise indiquant à Jahia que le code contenu est une ressource. Les ressources Jahia sont traitées de sorte à éviter les inclusions multiples de script identique, de plus ces ressources sont insérées dans la balise « header » comme le veulent les conventions de web.

### En-tête du site

#### Logo & Accroche

Ce logo a été conçu de sorte qu’il respecte le RGAA en plus de favoriser le référencement. La solution est lisible correctement par les lecteurs d’écrans (Lynx, VoiceOver).

D’après les spécifications envoyées par le client cette phrase est placée sous le titre du site, et est limitée en taille à une ligne. Cette notion subjective est dépendante des « dimensions » du site et est laissée à l’appréciation du client.

Pour simplifier l’insertion du logo et de l’accroche, un composant est chargé de remplir ces deux tâches. Ainsi, la vue permettant d’afficher le logo contient suffisamment d’information pour éviter la création d’un nouveau module.

#### Bandeau de navigation

Ce slider permet de mettre en évidence les 5 sites satellites de la Région Lorraine (« Entreprenez en Lorraine », « mylorraine.fr », « Le site officiel du tourisme en Lorraine », « Eureka Lorraine », « INFFOLOR »). Un existant étant déjà présent, quelques modifications ont été demandées.

##### Version 1 – Amélioration de l’existant basé sur ContentFlow

|  |  |
| --- | --- |
|  |  |
| Figure - Slider sans focus | Figure - Slider avec focus |

Il y a différentes interactions au niveau de ce slider. Le premier est une animation sur l’ensemble des items, à l’état « zéro » la représentation est sous forme pyramidale, à l’état « un » tous les items sont au même niveau. De plus, au survol d’un des items présents sur le bandeau, une zone est mise à jour avec le lien survolé.

L’existant se base sur le framework ContentFlow, dont il est possible de modifier la vue de façon basique. Du fait qu’il ne contient pas tous les événements existant en JavaScript, il a fallu lui rajouter une couche indépendante afin de créer l’animation lors de la prise du focus.

Suite à l’envoi de la maquette, la transition entre les deux états a été remarquée trop abrupte. Après étude du framework, la conclusion qui s’est imposée est qu’il n’est pas possible d’ajouter un effet de transition entre les deux états. Une des raisons est que le code du framework est offusqué. Une autre, est que le framework n’a pas été conçu pour avoir des transitions autres que sur la prise et la perte de focus d’un item. Or le comportement souhaité dépend de la position de la souris et non d’un clic souris. Une solution a découlé de ces divers éléments.

##### Version2 – Solution basée sur ContentFlow et le fonctionnement du dock Mac OS X

Le principe de l’animation au survol de la zone est gardé. Le premier état est identique à la Figure 10 - Slider sans focus. Le second état consiste en l’animation d’un ensemble d’items en fonction de la position de la souris, par rapport à la zone contenant le bandeau de navigation. Plus l’item est proche du curseur de la souris, plus il est agrandi.

|  |  |
| --- | --- |
|  | |
| Figure - Etat initial | |
|  |  |
| Figure - Le bandeau de navigation t1 | Figure - Le bandeau de navigation t2 |

Le comportement prévu dans le cas de la désactivation du JavaScript est décrit dans la figure ci-dessous. Ce système permet de garantir la conservation des données.

Figure - Fonctionnement avec et sans JavaScript

Ce dock est également utilisé dans la partie Dock partenaire, avec une vue légèrement différente. Il a donc été mis sous la forme d’un plug-in jQuery. Ce dernier est chargé de calculer la taille et de l’appliquer à l’élément courant, le reste étant géré en CSS.

### Barre permanente

Dans le cahier des charges figurent certaines exigences telles que la présence de menus déroulants nommés « espace privé » et « entrée par cible ». Ces menus sont présents dans une barre qui a une position fixe à l’écran. Elle est semblable à une barre des tâches et offre des accès rapides à certaines parties du site (page de recherches, sites satellites, liens externes et internes, etc.). Cette barre a pour exigence d’être fonctionnelle avec une taille de fenêtre de 800x600. En raison de la grande quantité d’information présente dans cette zone, cette exigence a été revue pour une dimension de 1024x768.

Figure - Maquette de la barre

La barre est un des composants de la charte qu’il a fallu redimensionner afin qu’elle soit entièrement visible.

#### Problématique du respect du RGAA

##### Par rapport au JavaScript

L’un des critères du RGAA consiste à avoir les mêmes informations avec et sans le JavaScript en terme de texte. On peut aborder le JavaScript sous plusieurs angles.

###### « Développement parallèle »

On peut développer en parallèle le JavaScript et le CSS, dans ce cas le risque est pris de perdre des informations si le JavaScript est désactivé. Car on peut avoir tendance à fixer le CSS en fonction du premier état de l’animation. Voici des aperçus tirés d’un prototype.

|  |  |
| --- | --- |
| Macintosh HD:Users:Nico:Desktop:Capture d’écran 2012-05-05 à 03.58.45.png | Macintosh HD:Users:Nico:Desktop:Capture d’écran 2012-05-05 à 03.58.45.png |
| Figure - Information visible avec JavaScript | Figure - Information sans JavaScript |

###### « Développement procédural »

Une autre solution consiste à développer d’abord le CSS en s’arrangeant pour que toutes les informations soient correctement agencées et lisibles et seulement ensuite introduire le JavaScript. À ce niveau le JavaScript à un rôle de restructuration du contenu en intervenant au niveau du CSS (style) sur différentes propriétés (hauteur, largeur, couleur, etc.).

|  |  |
| --- | --- |
| Macintosh HD:Users:Nico:Desktop:Capture d’écran 2012-05-05 à 04.09.43.png | Macintosh HD:Users:Nico:Desktop:Capture d’écran 2012-05-05 à 04.16.21.png |
| Figure - Information avec JavaScript | Figure - Information sans JavaScript |

##### Par rapport au cahier des charges

Les fonctionnalités développées en JavaScript doivent donc être conçues soigneusement afin que les informations puissent être accessibles si le script est désactivé. En effet, si elles sont mal conçues elles seront invisibles à l’écran. En plus de poser des problèmes d’accessibilités, une mauvaise conception peut entraîner un mauvais référencement pour les moteurs de recherches tels que Google.

Afin d’éviter tout problème de ce genre, la solution adoptée consiste à utiliser un développement procédural. Dans l’application de ce principe à un menu déroulant, la mise en « menu déroulant » est entièrement faite en JavaScript (jQuery) ainsi s’il est désactivé, le CSS permettra à toutes les informations d’être visible sur la page. De cette manière, le CSS sera chargé de faire la mise en forme sans animation, et le JavaScript mettra en forme le contenu avec les animations qui conviennent.

#### Généralité sur l’« entrée par cible » et l’« espace privé »

Les menus figurants dans la barre permanente fixe ont l’obligation de s’ouvrir vers le haut, du fait même que cette barre a une position absolue en bas de l’écran.

Ces deux menus sont liés par un lien d’exclusion mutuelle. C'est-à-dire que si l’un des menus est visible, le second ne doit pas l’être. Afin de gérer au mieux les similitudes de ces deux menus et la relation qui les lie, un framework a été développé en tant qu’extension de jQuery, jQueryMenu.

Il est chargé de deux missions. La première est d’initialiser le composant. L’initialisation consiste à « plier le menu », puis attribuer la fonction d’animation. La première étape, dans le cas où le JavaScript est désactivé est ignorée ce qui permet de garder un rendu des différents éléments composant le menu.

#### Service en ligne et Entrée par cible

L’entrée par cible est une nouvelle vue ajoutée au module service en ligne. Cette vue contient deux éléments. Le premier est le menu déroulant. Le second est un libellé qui sert de raccourci vers le dernier lien activé. Ce lien est stocké dans un cookie qui est renseigné lors du clic sur un des liens composant le menu. Afin d’inciter l’utilisation de ce menu, il est par défaut ouvert tant que le cookie est vide.

Figure - Entrée par cible

De plus ce menu doit pouvoir contenir des éléments navigables et non navigables, pour ce faire on ajoute dans la définition du composant un booléen permettant d’identifier le lien non cliquable.

Figure - Fenêtre de conception d'un service en ligne non navigable (lien non navigable)

Chaque entrée du menu mène vers un service particulier ; entre autres vers des formulaires qu’il a fallu uniformiser avec la charte correspondante. Cette uniformisation a servi de prétexte à rendre les différents formulaires plus accessibles et utilisant du JavaScript non intrusif comme décrit dans la partie précédente. Une difficulté imprévue s’est révélée au moment d’uniformiser un formulaire utilisant la bibliothèque JavaScript Prototype. Cette bibliothèque rentre en conflit avec la version de jQuery mise à disposition par Jahia. Ce conflit empêche l’exécution du JavaScript dans la page utilisant ce formulaire c’est pourquoi il a fallu adapter le code Prototype à jQuery. Les sliders de Prototype sont remplacés par les sliders de jQuery UI, celui-ci tient bien sûr compte des problèmes de désactivation du JavaScript.

|  |  |
| --- | --- |
|  |  |
| Figure - Ancien formulaire | Figure - Nouveau formulaire |
|  |  |
| Figure - Formulaire avec JavaScript désactivé | |

#### Espace privé

Ce menu est un simple menu déroulant. Bien qu’il fasse partie de l’existant, sa nouvelle position, dans la barre permanente (originellement dans l’en-tête), oblige à refaire le menu afin qu’il s’ouvre dans le bon sens.

Figure - Espace privé

#### Dock partenaire

##### Comportement déduit de la maquette

Au vu de l’espace alloué à ce slider (au vu de la taille de la barre), le comportement est différent. Pour celui-ci, seul l’item courant est animé. Il grandit sensiblement par rapport aux autres. De la même manière que pour le premier slider, un événement est ajouté en guise de couche supplémentaire au framework, afin de créer cette animation.

##### Comportement attendu du client

Le comportant, souhaité par le client, est similaire au comportement du dock Mac OS X. C'est-à-dire que les items du dock grandissent proportionnellement à la distance qui les sépare du curseur de la souris.

###### Ce que nous avons compris de la maquette

|  |  |
| --- | --- |
|  |  |
| Figure - Slider sans focus | Figure - Slider avec focus sur un item |

###### Ce qui est voulu

|  |  |
| --- | --- |
|  |  |
| Figure - Slider sans focus | Figure - Slider avec focus sur un item |

##### Éviter les redondances d’informations entre les deux sliders

De plus, les données de ce slider sont identiques à celles figurant dans la têtière. Afin d’éviter la redondance d’information, un composant particulier a été développé. Celui-ci permet de sélectionner un autre élément figurant dans la page et de récupérer le modèle associé à cet élément ; ceci est rendu possible grâce à l’héritage d’un composant Jahia. Ce système permet de manipuler, entre autres la vue, tout en partageant le même modèle que le composant précédemment lié. Ainsi, modifier l’élément de la têtière, modifie également celui figurant dans la barre permanente.

#### Outil publicité

Cet outil a les mêmes contraintes que les menus, à savoir s’ouvrir vers le haut. Il met donc aussi en œuvre le plug-in jQueryMenu. De plus, une période de validité contraint l’affichage de la publicité à cinq secondes, elle reste néanmoins accessible via un bouton dans le cas où la publicité est hors date de validité.

|  |  |
| --- | --- |
|  |  |
| Figure – Publicité non déployée | Figure - Publicité déployée |

### La cartographie des actions régionales

L’enjeu de la modification de ce module est de définir un style par défaut pour tous les formulaires du portail des Lorrains et de donner un exemple de formulaire correctement conçu permettant une bonne accessibilité.

#### L’existant

La cartographie est un module qui permet de représenter sous forme de carte la répartition des diverses actions engagées par la région. Ce module est accompagné d’un formulaire de recherche servant de filtre qui permet de restreindre les résultats selon les années, et divers axes politiques.

#### Le souhait de la région

La région souhaite ajouter un élément dans le formulaire afin de pouvoir effectuer une recherche par mots clefs. Ces mots clefs permettront de restreindre les résultats sur le libellé, le territoire (ville), le département, ainsi que le nom du fichier PDF associé, contenant l’action entreprise. Cette mise à jour s’accompagne du remplacement de la carte existante ainsi que de la vue du module.

#### Le formulaire

Certaines modifications ont été apportées afin d’améliorer l’accessibilité. En effet, certains éléments du formulaire ont été détournés de leur fonction première. Par exemple, le bouton validant les données avait une méthode « onclick »[[9]](#footnote-9) associée, il n’était donc pas dépendant du formulaire. Ceci ayant pour conséquence de poster des données pouvant être erronées ou encore désactiver les fonctionnalités inhérentes au formulaire[[10]](#footnote-10) à savoir l’envoie du formulaire sans passer par le JavaScript.

L’action sur ce bouton a donc été déplacée dans l’appel à la validation du formulaire par le biais de l’attribut « onsubmit » de la balise « form ». Cependant, la validation de ce formulaire entraîne aussi le rechargement de la page, ce qui n’est pas souhaité, car des actions en AJAX sont associées à ce formulaire. La particularité du « onsubmit » est d’attendre le retour d’un booléen qui indique si l’envoi doit être exécuté ou non. Ce booléen vaut « vraie » par défaut, ce qui a pour conséquence de poster le formulaire. En rajoutant un « return false », le formulaire n’est pas posté, et l’AJAX est exécuté.

En procédant de cette manière, on ouvre de nouvelles possibilités à ce formulaire, par exemple l’envoie des données par l’appui sur la touche « Entrée ».

De plus, le formulaire a été restructuré, des balises « label », « fieldset », « legend » ont été rajoutées afin d’améliorer la structure du formulaire et son accessibilité.

Le travail sur ce formulaire permet de définir un style par défaut pour les éléments de formulaire qui seront utilisés sur le portail.

##### La balise « label »

Cette balise permet de décrire un champ du formulaire, elle est indispensable à la compréhension d’un formulaire pour de nombreuses personnes employant des méthodes de navigation alternatives. Sans elle, ces personnes seront dans l'impossibilité de comprendre le rôle de chaque champ du formulaire, et donc de le remplir.

De plus, l’un des avantages apportés par cette balise est de pouvoir offrir un accès plus rapide au champ qu’elle qualifie. Par exemple, un clic sur le label permettra de mettre le focus sur une zone de saisie ou de cocher un bouton « radio » ou « checkbox ».

##### Les balises « fieldset » et « legend »

La première balise sert à définir une zone particulière d’un formulaire, elle sert à regrouper les champs d’un formulaire par thème. La seconde balise permet de donner un titre à la zone du défini par « fieldset ».

#### Modification de la requête SQL

Cette requête est écrite avec le langage JQOM, c’est un langage qui permet d’écrire une requête en utilisant des tags. La modification de cette requête a été délicate pour la partie concernant la recherche dans le nom du fichier joint à l’action régionale. En effet, le système de gestion de données de Jahia permet de définir des weakreferences[[11]](#footnote-11). Ce système empêche d’accéder directement aux données relatives à l’objet possédant ces attributs. Une fois ce système de gestion assimilé, une parade a été trouvée ; celle de passer par une requête récupérant la liste des fichiers correspondants aux mots clefs, puis d’injecter ces résultats dans la requête principale.

#### La nouvelle carte

La cartographie possède certaines animations, en effet au survol d’une région, la région courante est mise en surbrillance, ce système est géré via les balises « map » et « area » qui permettent de délimiter diverses zones. Au survol d’une région, une zone flottante s’ouvre avec les actions dépendantes du territoire sélectionné. Cette zone a été prévue afin qu’elle puisse être déplacée sur la page, conformément à la demande du client. Cette interaction est gérée par l’extension de jQuery pour les interfaces : jQuery UI.

|  |  |
| --- | --- |
|  |  |
| Figure - Zone flottante en position par défaut | Figure - Zone flottante dans une autre position |

Plusieurs échanges avec le client ont permis de mettre en évidence les défauts de la charte graphique. La troisième version de la carte, plus complète et mieux faite, a levé les problèmes soulevés dans les premières versions ;entre autres la non-exhaustivité de la carte : carte des régions survolées incomplètes, police illisible, et charge de travail importante pour obtenir un résultat conforme aux attentes.

### La newsletter

#### Existant

Cette fonctionnalité du portail des Lorrains est gérée par le système de portlet depuis la première version. Celle-ci se décompose en trois vues qui ont été mise à jour en fonction de la charte graphique des formulaires. La première est celle qui permet à un utilisateur de s’inscrire la newsletter.

Figure - Inscription à la newsletter

La seconde est une interface d’administration, qui permet à l’administrateur de configurer certains messages ou filtre.

Figure - Administration de la newsletter

La dernière permet à un utilisateur de choisir ses abonnements.

Figure - Choix des abonnements

#### La demande

À l’occasion de la mise à jour du site, la région a souhaité mettre en place un formulaire d’accès rapide à l’inscription de la newsletter.

Figure - Formulaire d'accès rapide à la newsletter

Afin d’avoir une navigation agréable, le formulaire de la Figure 38 doit mener au formulaire de la Figure 37. Afin d’avoir ce comportement, plusieurs hypothèses ont été envisagées. La première est de copier la valeur de l’attribut « action » du formulaire dans le nouveau formulaire. Le problème de cette méthode est que le champ « action » du formulaire d’inscription est généré par la portlet ce qui la rend non viable.

Dans un premier temps, le nouveau formulaire va « pointer » vers une action (similaire à une action Struts) qui permet la vérification de formulaire ou le traitement des données avant d’exécuter une redirection vers la page de destination.

Dans un second temps, il faut faire communiquer Jahia et les portlets. Pour réaliser cette communication, Jahia met à disposition un objet qui est partagé en session entre le CMS et les portlets.

Pour finir, une fois cet objet partagé, il faut récupérer les données à partir de la portlet et définir le bon affichage de formulaire c’est-à-dire le - Choix des abonnements.

Figure - Communication formulaire / portlet

### Recherche Synomia

De la même manière, une mise à jour de la vue est demandée. Cette mise à jour permet de corriger un défaut de conception du formulaire qui peut occasionner des défauts d’agencement sur certains navigateurs. À cette occasion, un découpage plus minutieux a été fait ; l’image du bouton a été extraite pour être placée sur le bouton de validation de formulaire, la forme de la zone de saisie est stylisée grâce au CSS.

|  |  |
| --- | --- |
|  |  |
| Figure – Ancien formulaire résultat attendu | Figure – Ancien formulaire résultat obtenu sur certains navigateurs |
|  | |
| Figure - Nouveau formulaire résultat uniforme sur les principaux navigateurs[[12]](#footnote-12) | |

#### Différentes manières de poster un formulaire

Il existe deux méthodes pour soumettre un formulaire de manière non intrusive. La méthode la plus répandue est d’utiliser le type « submit » de la balise « input ». La deuxième méthode, un peu moins connue, est d’utiliser le type « image » de la balise « input ». Ce type permet de soumettre un formulaire, au même titre que le « submit », mais il permet de définir l’apparence du bouton grâce à une image passée en attribut à cette balise. On peut aussi lui définir un attribut « alt » au cas où l’image n’est pas accessible.

|  |  |
| --- | --- |
|  |  |
| Figure – Aperçu du type « submit » | Figure – Aperçu du type « image » |

### Les actualités

Le module actualité se décline sous plusieurs composants, les actualités à la une et les remontées d’actualités sont concernées par les changements de charte graphique.

#### Les actualités à la une

Les actualités à la une permettent aux clients d’indiquer à l’aide d’une case à cocher si l’actualité courante doit figurer dans la liste des actualités importantes. Le contenu des articles à la une est mis en évidence par le biais de l’utilisation d’un dégradé permettant de mieux distinguer le titre « Actualités à la une » de la liste d’actualités à afficher.

Figure - Nouvelle vue de l’actualité à la une

#### Les dernières actualités

Ce composant permet de récupérer, chronologiquement, les dernières actualités, au maximum « n ». La vue change radicalement, en effet, dans la nouvelle version, une image descriptive de l’actualité est affichée, de même une description apparaît.

|  |  |
| --- | --- |
|  |  |
| Figure – Ancienne vue des « n » dernières actualités | Figure – Nouvelle vue des « n » dernières actualités |

### Les Articles

Un article se définit par un titre ainsi qu’un contenu riche que le client peut remplir à son gré en images, liens, textes, titres, etc.

La zone de contenu, qui est pleinement éditable par le client, est encadrée par un léger dégradé, qui permet de mieux distinguer le titre principal, du contenu informatif.

Figure - Nouvelle vue des articles

### Galerie Photo & Vidéo

Les modifications apportées à ce module se déclinent par plusieurs modifications. La première a été réalisée par Alexandre.

|  |  |
| --- | --- |
|  |  |
| Figure – Ancienne actualité vidéo | Figure – Nouvelle actualité vidéo |

La modification concernait le fonctionnement de l’actualité vidéo. La seconde modification concernait l’aspect de la galerie photo & vidéo.

|  |  |
| --- | --- |
|  |  |
| Figure – Ancienne galerie | Figure – Nouvelle galerie |

## Les Tests

Les premiers tests exécutés ont été réalisés « à la volée » lors du développement. Ces tests sont qualifiés de tests unitaires. Ceux-ci ne sont pas suffisants pour s’assurer du bon fonctionnement d’une future livraison. C’est pour cela que des tests sont réalisés sur d’autres environnements, dont l’environnement dit de développement et d’intégration. La plateforme de développement sert à faire une première série de tests, visant à vérifier le fonctionnement de chaque module séparément afin de vérifier qu’ils correspondent à la demande.

À partir de la plateforme de développement, un manuel d’installation est rédigé afin de guider le client lors de la procédure de mise à jour.

Sur la plateforme d’intégration, la procédure de déploiement est réalisée entièrement en suivant le manuel d’installation en pièce jointe de la livraison. Sur cette plateforme, on réalise la procédure d’installation que le client a à réaliser. La finalité est de tester tout ce qui est livré au client.

### En développement

Les données, initialement présentes sur la plateforme, sont issues de la préproduction[[13]](#footnote-13).

Les tests réalisés à cette occasion ont permis de valider une première fois le travail réalisé lors du développement. Durant cette période de tests, certains problèmes ont été levés. Problème de compatibilité de la barre permanente avec la résolution de 800x600, d'affichage d’ascenseur horizontal avec les résolutions de 800x600 et 1024x768. Reprise des développements à cause d’oublis d’exigences ou de fonctionnalités qui n’ont pas passé les tests. À titre d’exemple, les tests unitaires sur le module de publicité ont été réussis, mais n’ont pas tous été validés sur la plateforme de développement. En effet, la publicité ne s’affichait pas alors que la date de validité était vérifiée. Ceci était dû à un problème de conversion de la date du jour en date compréhensible par la JSTL[[14]](#footnote-14).

### En intégration

Le passage en intégration a permis de tester plusieurs éléments. Tout d’abord, tester la procédure de mise à jour que la région doit suivre afin de passer à la nouvelle version du site. La seconde a permis de tester la non-régression des modules existants ou de valider le bon fonctionnement des nouveaux modules. Mais aussi de signaler quelques points importants qui peuvent poser problème, chez le client, avec le changement de version. Par exemple des modules mal utilisés ou encore des problèmes inhérents à la combinaison de vidéos et à la barre permanente. En effet, certains sites de partage vidéos peuvent recouvrir partiellement la barre.

## Référencement et Accessibilité

### Référencement SEO

La région lorraine a décidé d’améliorer son référencement par les moteurs de recherche afin d’augmenter la fréquentation du site. Un audit SEO[[15]](#footnote-15) et RGAA a été effectué afin de déterminer quels points sont à améliorer. De cet audit sont ressortis les points suivants.

#### Absence de Page d’erreur 404

Jahia étant un CMS pouvant contenir plusieurs sites, il possède des pages d’erreurs par défaut. La région a souhaité avoir une page d’erreur 404 personnalisée proposant un contenu indicatif ou pouvant avoir un intérêt pour l’internaute.

On peut indiquer à Jahia où rechercher ces pages d’erreurs personnalisées afin de les définir comme page par défaut du site auquel s’applique le template.

Le principe consiste à se servir d’une page comme source de données. Lorsque Jahia va détecter une erreur 404, la page personnalisée sera affichée ; celle-ci ira rechercher le contenu à afficher dans la page prévue à cet effet.

Cette manière de procéder permet de conserver le code d’erreur 404, tout en proposant un contenu dynamique pour l’internaute et pour l’administrateur. Ce dernier aura ainsi l’occasion de personnaliser à souhait le contenu de la page d’erreur avec le contenu qu’il souhaite.

D’un point technique, les pages d’erreurs sont de simples pages JSP isolées du fonctionnement classique de Jahia. Les fonctions de Jahia ne peuvent donc pas être utilisées dans ce type de page. La récupération des données s’effectue par le biais d’une scriptlet[[16]](#footnote-16), qui ouvre une connexion vers la page contenant la source de données, qui récupère le flux et l’insère dans la page.

Cette méthode n’est pourtant pas encore satisfaisante. En effet, à partir du moment où l’on définit une page d’erreur personnalisée, celle-ci est automatiquement appelée par Jahia ; dans l’éventualité où la page « source de données » est trouvable, une erreur 404 va être retourné. Ce qui va provoquer un début d’appel récursif vers la page d’erreur 404.

Jahia acceptant le passage de paramètre en GET, la récursivité peut être évité en effectuant un test sur un paramètre unique passé à la page « source de données ». Si ce paramètre est « null » alors c’est le premier appel à la page sinon c’est le deuxième appel. En effet au premier appel, le paramètre unique n’est pas encore transmis à la page « source de données ». De plus dans le cas où la « source de données » n’existe pas – pas encore – la page d’erreur 404 par défaut de Jahia est affichée.

#### Référencement de page « technique »

Actuellement, le Portail des Lorrains est entièrement référencé par les moteurs de recherche, y compris le code JavaScript, CSS, RSS, JSON, mais aussi les pages servant de source de données (similaire à l’erreur 404).

Pour éviter ce référencement inutile, nous avons défini un fichier robots.txt qui indique quelles pages peuvent ou ne peuvent pas être référencées. De plus, ce fichier permet d’indiquer le fichier sitemap.xml qui contient les données nécessaires aux moteurs de recherche pour déterminer quelles pages doivent être réindexées en fonction de leur contenu.

Afin de tester le fichier robots.txt, nous avons utilisé le logiciel Screaming Frog SEO Spider Tool. Ce logiciel permet de vérifier l’indexation des pages, ainsi que de visualiser les principaux éléments utiliser pour le référencement comme les titres, description de la métadonnée, etc.

#### Conflit « alt » et « title »

Il est également mis en évidence un conflit entre ces deux attributs, après analyse la solution suivante a été approuvée. Toute image définie dans un composant se voit associer un texte alternatif, afin que l’utilisateur ait la possibilité de donner une alternative la plus pertinente possible.

##### Image

Dans ce cas, le « title » peut être vide ou renseigner, mais le « alt » doit être renseigné.

##### Image lien

Dans ce cas, le « title » de la balise « a » doit être plus grand l’attribut « alt » de la balise image afin de préciser l’intérêt du lien. Dans ce cas le contenu textuel est déterminé par l’attribut « alt »

<a href="http://www.google.fr" title="Aller vers google">

<img src="https://www.google.fr/images/srpr/logo3w.png" alt="Google"/>

</a>

##### Texte lien

Dans ce cas, le « title » de la balise « a » doit être plus grand que sa valeur. Afin de le préciser par exemple ceci est valide :

<a href="http://www.google.fr" title="Aller vers google">google</a>

De plus, le « title » est renseigné intelligemment. En effet, Jahia permet d’indiquer des métadonnées permettant de décrire un composant. Aussi cette valeur est utilisée dans le « title » des liens.

### Accessibilité

# Conclusion

# Table des Annexes

Annexe i - Commentaire jour à jour A

Annexe ii - Maquette du portail des Lorrains v2 K

Annexe iii - Autre L

# Annexe

## Annexe - Commentaire jour à jour

| Date | Commentaire |
| --- | --- |
| 19-mars-12 | Lecture des documents concernant le fonctionnement d’Atos (projet, etc.) installation poste |
| 20-mars-12 | Installation sur le poste des logiciels (Jahia, base de données, SVN, etc.) |
| 21-mars-12 | Suite de l’installation des postes (Jahia, base de données, SVN, etc.) |
| 22-mars-12 | Premier jour de formation Jahia |
| 23-mars-12 | Deuxième jour de formation Jahia |
| 26-mars-12 | Révision du fonctionnement des tests Mise au point de la structuration des projets Correction mineure sur le portail des Lorrains  Prétest sur machine de développement |
| 27-mars-12 | Test sur plateforme d'intégration |
| 28-mars-12 | Mise en forme des tests (complément d'information sur les explorateurs internet IE, Firefox).  Recherche sur ReadSpeaker (transmission des recherches effectuées dans la fin de journée)  Déploiement des modifications sur plateforme d’intégration |
| 29-mars-12 | Reprise des recherches sur ReadSpeaker.  Apprentissage de l’architecture de Jahia (où sont stockées les archives war) 1er jet ReadSpeaker |
| 30-mars-12 | Correction de documentation à usage du client (SFD, tests références et lot d'évolution mineure)  rédaction documentation générale ReadSpeaker |
| 02-avr-12 | Reprise des documents client de la fiche Mantis 205 et 211.  Livraison et analyse du festival des musiques lycéennes. Rédaction du questionnaire pour client ReadSpeaker |
| 03-avr-12 | Rédaction de SFD ReadSpeaker et mise à jour SFD.  Création d’une branche « festival des musiques lycéennes » et ajout d'une vue FML |
| 04-avr-12 | Développement de « article FML » avec l'aide de la maquette du client |
| 05-avr-12 | Confirmation du développement début de rédaction |
| 06-avr-12 |  |
| 09-avr-12 |  |
| 10-avr-12 | Rédaction des documents.  Vérification sur « crlo - article » apport de modification pour limiter les cas extrêmes et chaos |
| 11-avr-12 | Finalisation du SFD, DAF, BLI pour tests en intégration. Modification sur une demande du client suite à l'envoi de la maquette |
| 12-avr-12 | Correction des spécifications (quiproquo sur le fonctionnement souhaité des votes). Réévaluation du DAF, SFD, de la fiche de test ainsi que la manière de cacher le compteur. Affichage du compteur uniquement pour la région lorraine |
| 13-avr-12 | Changement dans la manière de présenter les articles votables, mise en œuvre d'une liste qui arrange des composants sous 2 colonnes |
| 16-avr-12 | Découverte d'un bogue d'affichage lié apparemment au cache. Pour vérification déploiement sur une version plus récente de Jahia. |
| 17-avr-12 | Découverte d'un bogue d'affichage lié apparemment au cache. Pour vérification déploiement sur une version plus récente de Jahia. |
| 18-avr-12 | Travail sur le bandeau mis au point de plusieurs solutions à proposer au client basé sur ContentFlow. |
| 19-avr-12 | Débogage pour équivalence entre navigateurs du nouveau bandeau |
| 20-avr-12 | Intégration du bandeau dans Jahia. Ajout de questions au client à propos du drag and drop, et sur le menu déroulant et la zone fixe |
| 23-avr-12 | Suite d'analyse du CCTP, test sur le menu inférieur et réflexion sur la barre permanente. Début d’intégration |
| 24-avr-12 | Intégration du menu déroulant ascendant. Amélioration selon la maquette du client. Fin des développements avec les données du moment, seules les fonctionnalités supposées sont OK |
| 25-avr-12 | Correction à des problèmes sur FML, correction de bogue sur le site Eureka |
| 26-avr-12 | Correction bogue sur Eureka. Impossibilité de tester le fonctionnement à cause des restrictions de sécurité sur les plateformes |
| 27-avr-12 | Rectification de bogue sur la barre permanente. Amélioration du comportement du menu déroulant et modification de la vue de l'article FML |
| 30-avr-12 |  |
| 01-mai-12 |  |
| 02-mai-12 | Résolution de divers problèmes sur la vue d'un article simple. Rédaction document pour évolution PDL |
| 03-mai-12 | Réflexion sur les problèmes d'accessibilité sur les menus déroulants + maj SFG et SFD pour la v2 (SFG terminé, SFD en cours) |
| 04-mai-12 | Fin de rédaction SFD. Vérification des normes d’accessibilités sur certains modules notamment pour les scripts |
| 07-mai-12 | Travail sur le slider inférieur. Étude de l’accessibilité des sliders avec désactivation JavaScript |
| 08-mai-12 |  |
| 09-mai-12 | Correction du bogue d'animation sous Chrome et d’Internet Explorer. Création d'un composant pour le slider fixe.  Travail sur la publicité : ajout des bornes de validités et d'affichage et utilisation du plug-in jQueryMenu pour l'affichage de la publicité.  Ajout du message au survol d'un lien non navigable.  Ajout mis à jour de document pour donner suite aux réponses du client. |
| 10-mai-12 | Travail sur la cartographie. Document SFG, mise à jour des questions pour le client |
| 11-mai-12 | Proposition de plusieurs types de recherche sur la cartographie.  Réflexion sur la recherche du nom de fichier |
| 14-mai-12 | Requête sur cartographie avec recherche par mots clefs. Fonctionne avec le libelle, territoire, ainsi que le nom du fichier. Ajout de la fonctionnalité draggable. Réunion de début de projet sur le PDL v2 |
| 15-mai-12 | Découpage de la charte graphique pour test sur la barre permanente fixe |
| 16-mai-12 | Découpage de la charte graphique pour test sur la barre permanente fixe complément avec Loïc pour l'uniformisation du code. |
| 17-mai-12 |  |
| 18-mai-12 |  |
| 21-mai-12 | Finalisation de l'intégration de la charte dans la barre fixe finalisation du développement. Conception d'une maquette statique |
| 22-mai-12 | Intégration dans la maquette de l'en-tête et de la barre fixe. Début d'uniformisation de la charte |
| 23-mai-12 | Maquettage du zoom et de l’actualité |
| 24-mai-12 | Travail sur la carte, refactoring des styles communs au site, découverte de bogue sous Internet Explorer |
| 25-mai-12 | Finition sur le maquettage central. Formation procédé tests |
| 28-mai-12 |  |
| 29-mai-12 | La « formation » Photoshop a permis de faire le point avec le formateur sur ce qu'il aurait fallu faire lors du découpage (ombre, transparence, taille, etc.) |
| 30-mai-12 | Ajout de la page « entreprendre » à la maquette. Apport de corrections sur le CSS et intégration de pdl\_jah\_1.5.0 dans le trunk et dans pdl\_jah\_2.0.0 |
| 31-mai-12 | Ajout de la remontée d'actualités. Découpage des boutons de la galerie, modification du comportement de la publicité et modification sur l'article |
| 01-juin-12 | Réunion avec le client pour la v2, préparation de la maquette pour la réunion affectation du FML avec demande d’évolution mineure |
| 04-juin-12 | Travail sur le FML mis en forme du bandeau de navigation inférieur sous forme de dock jquerisé |
| 05-juin-12 | Livraison du FML 1.5.4. Réunion d'avancement du PDL. Ajout de détail CSS. Intégration à Jahia des composants : accroche, logo, bandeau de navigation (haut, bas), publicité |
| 06-juin-12 | Vérification des fonctionnalités des modules intégrés à Jahia. Intégration des composants : « publicité », « service en ligne », « espace privé ». Difficulté à intégrer les actualités à la une => problème de compréhension du jmix\_list. |
| 07-juin-12 | Intégration à Jahia des composants zoom, article, actualité, RSS |
| 08-juin-12 | Révision de certains modules pour zoom, cartographie, logo et de l’accroche pour que le client ait plus de facilité à intégrer |
| 11-juin-12 |  |
| 12-juin-12 |  |
| 13-juin-12 |  |
| 14-juin-12 |  |
| 15-juin-12 |  |
| 18-juin-12 |  |
| 19-juin-12 |  |
| 20-juin-12 |  |
| 21-juin-12 |  |
| 22-juin-12 |  |
| 25-juin-12 |  |
| 26-juin-12 |  |
| 27-juin-12 |  |
| 28-juin-12 |  |
| 29-juin-12 |  |
| 02-juil-12 |  |
| 03-juil-12 |  |
| 04-juil-12 |  |
| 05-juil-12 |  |
| 06-juil-12 |  |
| 09-juil-12 |  |
| 10-juil-12 |  |
| 11-juil-12 |  |
| 12-juil-12 |  |
| 13-juil-12 |  |
| 16-juil-12 |  |
| 17-juil-12 |  |
| 18-juil-12 |  |
| 19-juil-12 |  |
| 20-juil-12 |  |
| 23-juil-12 |  |
| 24-juil-12 |  |
| 25-juil-12 |  |
| 26-juil-12 |  |
| 27-juil-12 |  |
| 30-juil-12 |  |
| 31-juil-12 |  |
| 01-août-12 |  |
| 02-août-12 |  |
| 03-août-12 |  |
| 06-août-12 |  |
| 07-août-12 |  |
| 08-août-12 |  |
| 09-août-12 |  |
| 10-août-12 |  |
| 13-août-12 |  |
| 14-août-12 |  |
| 15-août-12 |  |
| 16-août-12 |  |
| 17-août-12 |  |
| 20-août-12 |  |
| 21-août-12 |  |
| 22-août-12 |  |
| 23-août-12 |  |
| 24-août-12 |  |
| 27-août-12 |  |
| 28-août-12 |  |
| 29-août-12 |  |
| 30-août-12 |  |
| 31-août-12 |  |
| 03-sept-12 |  |
| 04-sept-12 |  |
| 05-sept-12 |  |
| 06-sept-12 |  |
| 07-sept-12 |  |
| 10-sept-12 |  |
| 11-sept-12 |  |
| 12-sept-12 |  |
| 13-sept-12 |  |
| 14-sept-12 |  |

Annexe - Maquette du portail des Lorrains v2

Annexe - Autre

Passer par insertion -> légende -> intitulé Annexe

# Table des matières

Remerciement 1

Sommaire 2

Table des illustrations 4

Introduction 7

Présentation d’Atos 8

Sujet 9

Cycle de vie d’un projet 10

Le Portail des Lorrains (PDL) 12

Présentation de Jahia 12

Développement 12

Syntaxe 13

Module 14

Template 14

Portlet 14

Les trois types de requêtes possibles sous Jahia 14

Le SQL 2 14

Le JQOM (Java Query Object Model) 14

Le XPATH 14

Travaux relatifs au Portail des Lorrains - version 1 15

FML 15

Conceptions — différentes solutions possibles : 16

Solution 1 16

Solution 2 17

Solution retenue 17

Problèmes rencontrés 17

Les remarques du client 17

Évolution en parallèle au développement du Portail des Lorrains Version 2 18

Version 2.0 du Portail des Lorrains 19

Gestion de projet - Déroulement du développement de la version 2 19

Outils utilisés 20

Problématique du découpage de certains éléments 20

Le patron du site 21

En-tête du site 21

Logo & Accroche 21

Bandeau de navigation 22

Version 1 – Amélioration de l’existant basé sur ContentFlow 22

Version2 – Solution basée sur ContentFlow et le fonctionnement du dock Mac OS X 22

Barre permanente 23

Problématique du respect du RGAA 24

Par rapport au JavaScript 24

« Développement parallèle » 24

« Développement procédural » 24

Par rapport au cahier des charges 24

Généralité sur l’« entrée par cible » et l’« espace privé » 25

Service en ligne et Entrée par cible 25

Espace privé 26

Dock partenaire 26

Comportement déduit de la maquette 26

Comportement attendu du client 26

Ce que nous avons compris de la maquette 26

Ce qui est voulu 27

Éviter les redondances d’informations entre les deux sliders 27

Outil publicité 27

La cartographie des actions régionales 27

L’existant 27

Le souhait de la région 27

Le formulaire 28

La balise « label » 28

Les balises « fieldset » et « legend » 29

Modification de la requête SQL 29

La nouvelle carte 29

La newsletter 29

Existant 29

La demande 30

Recherche Synomia 31

Différentes manières de poster un formulaire 31

Les actualités 32

Les actualités à la une 32

Les dernières actualités 32

Les Articles 32

Galerie Photo & Vidéo 33

Les Tests 33

En développement 33

En intégration 34

Référencement et Accessibilité 34

Référencement SEO 34

Absence de Page d’erreur 404 34

Référencement de page « technique » 35

Conflit « alt » et « title » 35

Image 35

Image lien 36

Texte lien 36

Accessibilité 36

Conclusion 37

Table des Annexes 38

Annexe A

Annexe i - Commentaire jour à jour A

Annexe ii - Maquette du portail des Lorrains v2 K

Annexe iii - Autre L

Table des matières A

1. JCR : Java Content Repository [↑](#footnote-ref-1)
2. Rédacteur du rapport : Nicolas Reitz [↑](#footnote-ref-2)
3. IE 8 et + [↑](#footnote-ref-3)
4. Firefox v10 et + [↑](#footnote-ref-4)
5. Format d’enregistrement par défaut de Photoshop [↑](#footnote-ref-5)
6. Red Green Blue Alpha ; la particule Alpha gère la transparence [↑](#footnote-ref-6)
7. HTC est l’extension d’un fichier qui permet la création d’un style CSS [↑](#footnote-ref-7)
8. Les hacks CSS sont prévus pour pallier aux déficiences de la gamme Internet Explorer. Le hack le plus répandu est « !important » [↑](#footnote-ref-8)
9. Représente du JavaScript intrusif (bloquant) [↑](#footnote-ref-9)
10. Un élément formulaire fonctionne avec le JavaScript désactivé [↑](#footnote-ref-10)
11. Les weakreferences sont assimilables à des pointeurs [↑](#footnote-ref-11)
12. Internet Explorer 7 et plus, Firefox, Opera, Chrome, Safari [↑](#footnote-ref-12)
13. Données que le client utilise sur sa plateforme de tests [↑](#footnote-ref-13)
14. JSTL: Java Standard Tag Library. Permets d’écrire des parties d’algorithme en utilisant une syntaxe XML (boucle, etc.). [↑](#footnote-ref-14)
15. Search engine optimization [↑](#footnote-ref-15)
16. Une scriptlet JSP est un bloc de code Java, inséré au sein d’une page JSP, délimité par des balises. [↑](#footnote-ref-16)