



**HAL**  
open science

## Mise en conformité d'un logiciel universitaire de publication de contenus Web à vocation publique aux normes d'accessibilité du Web

Jérôme Strippoli

### ► To cite this version:

Jérôme Strippoli. Mise en conformité d'un logiciel universitaire de publication de contenus Web à vocation publique aux normes d'accessibilité du Web. Web. 2010. dumas-00523961

**HAL Id: dumas-00523961**

**<https://dumas.ccsd.cnrs.fr/dumas-00523961v1>**

Submitted on 6 Oct 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**CONSERVATOIRE NATIONAL DES ARTS ET METIERS**  
**CENTRE REGIONAL RHÔNE-ALPES**  
**CENTRE D'ENSEIGNEMENT DE GRENOBLE**

---

**MEMOIRE**  
**présenté par Jérôme STRIPPOLI**  
**en vue d'obtenir**  
**LE DIPLÔME D'INGENIEUR C.N.A.M.**  
**en INFORMATIQUE**

---

**Mise en conformité d'un logiciel universitaire de  
publication de contenus Web à vocation publique aux  
normes d'accessibilité du Web**

**Soutenu le 30 juin 2010**

---

**JURY**

**Président :** M. Eric Gressier-Soudan

**Membres :** M. Jean-Pierre Giraudin  
M. André Plisson  
M. Mathias Voisin-Fradin  
M. Marc Vassoille  
M. Eric Giraudin

**Invités :** M. Gérard Rameau  
Mme. Sylvie Pesty





**CONSERVATOIRE NATIONAL DES ARTS ET METIERS**  
**CENTRE REGIONAL RHÔNE-ALPES**  
**CENTRE D'ENSEIGNEMENT DE GRENOBLE**

---

**MEMOIRE**  
présenté par Jérôme STRIPPOLI

en vue d'obtenir  
**LE DIPLÔME D'INGENIEUR C.N.A.M.**  
en INFORMATIQUE

---

**Mise en conformité d'un logiciel universitaire de  
publication de contenus Web à vocation publique aux  
normes d'accessibilité du Web**

**Soutenu le 30 juin 2010**

---

Les travaux relatifs à ce mémoire ont été effectués au sein du service informatique  
de l'IUT2 de Grenoble sous la direction de *Monsieur Marc Vassoille*



# Remerciements

Je remercie, en premier lieu, toute l'équipe du pôle informatique de l'IUT2 de Grenoble, Paul CHOMAT, Alain CLOCHET, Thérèse GENTILE, Philippe PEYRUT, Philippe HABERT, Sylvie PESTY, Léon RENAUD, Jérôme POGGI et Philippe FRECHOU, qui ont su si bien m'accueillir et m'entourer.

Je remercie tout particulièrement, Monsieur Marc VASSOILLE, membre du pôle informatique de l'IUT2 et tuteur de mon stage, pour m'avoir encadré, soutenu et avoir su se rendre aussi disponible lorsque j'avais besoin de lui.

Je remercie Monsieur Michel MUSSI, Président Directeur Général de DEPAGNE S.A.S, pour m'avoir permis de réaliser ce stage.

Je remercie enfin les membres du jury pour avoir accepté de participer à la soutenance de ce mémoire :

- Monsieur Eric GRESSIER-SOUDAN, Professeur au CNAM de Paris, et président de ce jury.
- Monsieur Jean-Pierre GIRAUDIN, Directeur Adjoint de l'IUT2 de Grenoble, Responsable pédagogique et scientifique du cycle ingénieur CNAM Informatique de Grenoble et Professeur au département STID de l'IUT2.
- Monsieur André PLISSON, Directeur du Centre d'enseignement CNAM de Grenoble.
- Monsieur Mathias VOISIN-FRADIN, Sous-directeur du Centre d'enseignement CNAM de Grenoble.
- Monsieur Eric GIRAUDIN, Responsable du service communication de l'IUT2 de Grenoble.
- Monsieur Gérard RAMEAU, Chef de projet X-TEK et formateur TICE.
- Madame Sylvie PESTY, Professeur d'informatique à l'IUT2.



# Table des matières

<b>REMERCIEMENTS</b>	<b>I</b>
<b>TABLE DES MATIERES</b>	<b>III</b>
<b>LISTE DES FIGURES</b>	<b>V</b>
<b>LISTE DES TABLEAUX</b>	<b>VI</b>
<b>LISTE DES ABREVIATIONS ET ACRONYMES</b>	<b>VII</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1. Contexte	1
1.2. Objectif du stage	2
1.3. Présentation de X-TEK	3
1.3.1. Les CMS	3
1.3.2. X-TEK	4
1.4. Plan du Mémoire	6
<b>2. RAPPEL SUR L'ACCESSIBILITE DU WEB</b>	<b>9</b>
2.1. Les normes	9
2.1.1. WCAG 1.0	10
2.1.2. WCAG 2.0	13
2.2. Les référentiels d'accessibilité de la DGME	16
2.2.1. RGAA 1.0	16
2.2.2. RGAA 2.2	17
2.3. Synthèse	17
<b>3. AUDIT DE L'ACCESSIBILITE DE X-TEK</b>	<b>19</b>
3.1. Mise en place d'une plateforme de test	20
3.1.1. Les différents modules disponibles	20
3.1.2. La structure de la plateforme de test	24
3.2. La recherche de logiciels de tests d'accessibilité	30
3.3. Analyse des résultats	35
3.3.1. La fusion des rapports	36
3.3.2. Les données	38
3.3.3. Analyse globale	40
3.3.4. Analyse des modules	43
3.4. Synthèse	56
3.4.1. Exploitation	56
3.4.2. Conclusions	59
3.5. Recommandations générales	60
3.6. Définition du plan de mise en conformité	61
3.7. Conclusion	62
<b>4. MISE EN CONFORMITE</b>	<b>65</b>
4.1. Étude du fonctionnement de X-TEK	65
4.1.1. Partie graphique	65
4.1.2. Code source	67
4.1.3. Patron de conception	68
4.2. Déroulement de la mise en conformité	70
4.3. Application des tests	70



## Table des matières

---

4.3.1.	Thématique Standard	72
4.3.2.	Thématique « Navigation »	76
4.3.3.	Thématique « Scripts »	81
4.3.4.	Thématique « Images »	84
4.3.5.	Thématique « Couleurs »	86
4.3.6.	Thématique « Présentation »	88
4.3.7.	Thématique « Structure »	95
4.3.8.	Thématique « Tableau »	99
4.3.9.	Thématique « Texte »	101
4.3.10.	Thématique « Formulaire »	104
4.3.11.	Thématique « Cadre »	106
4.3.12.	Thématique « Multimédia »	106
<b>4.4.</b>	<b>Recherche d'un nouvel éditeur de texte enrichi</b>	<b>110</b>
<b>4.5.</b>	<b>Présentation des résultats</b>	<b>111</b>
<b>4.6.</b>	<b>Synthèse</b>	<b>112</b>
<b>5.</b>	<b>CONCLUSION</b>	<b>113</b>
<b>5.1.</b>	<b>Rappel des objectifs</b>	<b>113</b>
<b>5.2.</b>	<b>Travail réalisé</b>	<b>113</b>
<b>5.3.</b>	<b>Perspectives</b>	<b>114</b>
<b>5.4.</b>	<b>Bilan personnel</b>	<b>115</b>
<b>6.</b>	<b>BIBLIOGRAPHIE</b>	<b>117</b>
<b>7.</b>	<b>ANNEXES</b>	<b>119</b>
<b>7.1.</b>	<b>Annexe 1 : Points de contrôle du WCAG 1.0</b>	<b>119</b>
<b>7.2.</b>	<b>Annexe 2 : Critères de succès du WCAG 2.0</b>	<b>125</b>
<b>7.3.</b>	<b>Annexe 3 : Outils testés</b>	<b>129</b>
7.3.1.	Outils testés et retenus	129
7.3.2.	Outils testés et non retenus	136

# Liste des figures

<i>Figure 1 : Architecture générale d'un CMS .....</i>	<i>4</i>
<i>Figure 2 : Formulaire d'édition d'une page .....</i>	<i>21</i>
<i>Figure 3 : Dossier liste.....</i>	<i>22</i>
<i>Figure 4 : Dossier menu.....</i>	<i>22</i>
<i>Figure 5 : Arborescence de la catégorie « Page menu onglets » .....</i>	<i>25</i>
<i>Figure 6 : Arborescence de la catégorie « Page menu hypermots » .....</i>	<i>27</i>
<i>Figure 7 : Arborescence de la catégorie « Page menu déroulant » .....</i>	<i>28</i>
<i>Figure 8 : Arborescence de la catégorie « Les modules » .....</i>	<i>29</i>
<i>Figure 9 : Répartition des erreurs manuelles et automatiques selon leur priorité par indices de page groupés par module .....</i>	<i>41</i>
<i>Figure 10 : Moyenne des erreurs manuelles et automatiques cumulées par priorité et par module.....</i>	<i>42</i>
<i>Figure 11 : Répartition des erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité .....</i>	<i>43</i>
<i>Figure 12 : Répartition des erreurs automatiques et manuelles communes au module administration sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité ..</i>	<i>44</i>
<i>Figure 13 : Répartition des erreurs automatiques et manuelles communes au module calendrier sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité .....</i>	<i>45</i>
<i>Figure 14 : Répartition des erreurs automatiques et manuelles communes au module « dossier liste » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité ..</i>	<i>46</i>
<i>Figure 15 : Répartition des erreurs automatiques et manuelles communes au module « dossier menu » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité ..</i>	<i>47</i>
<i>Figure 16 : Répartition des erreurs automatiques et manuelles communes au module « dossier une seule page » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité (page d'accueil exclu).....</i>	<i>48</i>
<i>Figure 17 : Répartition des erreurs pour les pages d'indice 53 et 54, par points de contrôle groupés par priorité .....</i>	<i>49</i>
<i>Figure 18 : Répartition des erreurs automatiques et manuelles communes au module formulaire sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité .....</i>	<i>50</i>
<i>Figure 19 : Répartition des erreurs automatiques et manuelles communes au module forum sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité .....</i>	<i>51</i>
<i>Figure 20 : Répartition des erreurs automatiques et manuelles communes au module messagerie sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité .....</i>	<i>52</i>
<i>Figure 21 : Répartition des erreurs automatiques et manuelles communes au module trombinoscope sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité ..</i>	<i>53</i>
<i>Figure 22 : Répartition des erreurs automatiques et manuelles communes au module « flux rss » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité .....</i>	<i>54</i>

## Liste des tableaux

---

<i>Figure 23 : Répartition des erreurs automatiques et manuelles communes au module « liste des utilisateurs » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité.....</i>	<i>55</i>
<i>Figure 24: Répartition des erreurs automatiques et manuelles communes au module « page web » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité...</i>	<i>56</i>
<i>Figure 25: Différents templates de X-TEK.....</i>	<i>66</i>
<i>Figure 26 : Arborescence des dossiers du code source de X-TEK.....</i>	<i>67</i>
<i>Figure 27 : Patron de conception MVC.....</i>	<i>69</i>
<i>Figure 28: Exemple de menu Hypermot.....</i>	<i>89</i>
<i>Figure 29 : Exemple de menu Onglet.....</i>	<i>89</i>
<i>Figure 30: Exemple de menu d'administration.....</i>	<i>90</i>
<i>Figure 31 : Structure de la page standard.....</i>	<i>91</i>
<i>Figure 32 : Nouvelle structure de page.....</i>	<i>92</i>
<i>Figure 33: Structure finale de la page.....</i>	<i>93</i>
<i>Figure 34 : Plan de la page.....</i>	<i>97</i>
<i>Figure 35: Fleur en Art ASCII.....</i>	<i>102</i>
<i>Figure 36: Inclusion d'éléments "object".....</i>	<i>109</i>

## Liste des tableaux

<i>Tableau 1 : Liste des pages testées issues de la page d'accueil et de la catégorie « Page menu onglets » _____</i>	<i>26</i>
<i>Tableau 2 : Liste des pages testées issues de la catégorie « Page menu hypermots » _____</i>	<i>27</i>
<i>Tableau 3 : Liste des pages testées issues de la catégorie « Page menu déroulant » _____</i>	<i>28</i>
<i>Tableau 4 : Liste des pages testées issues de la catégorie « Les modules » _____</i>	<i>30</i>
<i>Tableau 5 : Liste des pages testées issues de la partie administrative _____</i>	<i>30</i>
<i>Tableau 6 : Mise en relation des trois tests automatiques et du test manuel _____</i>	<i>35</i>
<i>Tableau 7 : Indices des pages testées _____</i>	<i>38</i>
<i>Tableau 8 : Répartition des erreurs automatiques et manuelles _____</i>	<i>39</i>
<i>Tableau 9 : Répartition des erreurs automatiques et manuelles par priorité _____</i>	<i>43</i>
<i>Tableau 10 : Part des erreurs communes dans l'ensemble des erreurs _____</i>	<i>57</i>
<i>Tableau 11:Part des erreurs communes automatiques et manuelles dans l'ensemble des erreurs automatiques et manuelles _____</i>	<i>58</i>
<i>Tableau 12:part des erreurs communes dans l'ensemble des erreurs des modules _____</i>	<i>59</i>

# Liste des abréviations et acronymes

ADAE :	Agence pour le Développement de l'Administration Electronique
ARIA :	Accessible Rich Internet Applications
CMS :	Content Management System
CSS :	Cascading Style Sheets
CTIC :	Centro Tecnológico de la Información y de la Comunicación
DGME :	Direction Générale de la Modernisation de l'Etat
DTD :	Document Type Definition
HTML :	Hypertext Markup Language
MVC :	Modèle Vue Contrôleur
PHP :	Hypertext Preprocessor
RGAA :	Référentiel Général d'Accessibilité pour les Administrations
RSS :	« Rich Site Summary » ou « Really Simple Syndication »
SGBD :	Système de Gestion de Bases de Données
SMIL :	Synchronized Multimedia Integration Language
UWEM :	Unified Web Evaluation Methodology
W3C :	World Wide Web Consortium
WAI :	Web Accessibility Inspector
WAI :	Web Accessibility Initiative
WCAG :	Web Content Accessibility Guidelines 1.0
WYSIWYG :	What You See Is What You Get
XHTML :	Extensible Hypertext Language

## Liste des abréviations et acronymes

---

# 1.Introduction

## 1.1. Contexte

Peu de temps après la démocratisation de l'Internet dans le milieu des années 90, le W3C<sup>1</sup> s'est intéressé à l'accessibilité du Web aux personnes souffrant de différents handicaps tant visuel, auditif que moteur, en créant le WCAG 1.0<sup>2</sup> (Web Content Accessibility Guidelines), passé en recommandation le 5 mai 1999. Ce document est devenu la norme de référence pour l'accessibilité du Web sur laquelle se sont reposées toutes les initiatives visant à rendre le Web accessible à travers le monde.

Le 11 février 2005, le parlement Français adopte une loi portant sur « l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées ». Il est défini dans cette loi, à l'article 47, que les services de communication publique en ligne des services de l'Etat, des collectivités territoriales et des établissements publics qui en dépendent, doivent être accessibles aux personnes handicapées. Un décret d'application devait suivre cette loi et imposer une mise en conformité des services en ligne dans un délai de 3 ans ainsi que des sanctions imposées en cas de non-respect de cette mise en accessibilité. Il devait aussi indiquer quel référentiel d'accessibilité serait utilisé.

En 2004, un référentiel d'accessibilité avait été initié en collaboration entre l'ADAE<sup>3</sup> et l'association BrailleNet. Ce document s'est retrouvé inadapté à l'usage et un nouveau référentiel a été initié en 2006 par la DGME<sup>4</sup>, le RGAA<sup>5</sup>, afin de se conformer aux normes internationales. Il devait servir de référentiel d'accessibilité sur lequel se reposerait le futur décret.

Il s'est alors posé le problème de l'accessibilité du générateur de plateformes X-TEK, développé par l'IUT2 de GRENOBLE. X-TEK a servi à générer bon nombre de sites

---

<sup>1</sup> W3C : World Wide Web Consortium

<sup>2</sup> WCAG 1.0 : Directives d'accessibilité des contenus Web

<sup>3</sup> ADAE : Agence pour le Développement de l'Administration Electronique

<sup>4</sup> DGME : Direction Générale de la Modernisation de l'Etat

<sup>5</sup> RGAA : Référentiel Général d'Accessibilité pour les Administrations

institutionnels, il rentre ainsi dans le cadre de la loi de 2005 et se doit donc de respecter les normes d'accessibilité. Le service informatique de l'IUT2 m'a proposé comme sujet de stage la mise en conformité de X-TEK aux normes de l'accessibilité du Web. Dans un premier temps, j'ai effectué, dans le cadre de mon épreuve test, une étude sur le thème de « La problématique de l'accessibilité numérique pour un logiciel universitaire de publication de contenus Web à vocation publique » ; épreuve test soutenue en juin 2008.

Entre ma soutenance de 2008 et le début de mon stage en février 2009, le monde de l'accessibilité a sensiblement évolué. La version définitive du RGAA 1.0 est parue le 24 novembre 2008, suivie peu de temps après par le WCAG 2.0. C'est dans ce contexte changeant et qui changera encore que ce stage s'est déroulé.

### **1.2. Objectif du stage**

Ce stage avait pour but la mise en conformité de X-TEK par rapport au RGAA 1.0. Il nous était aussi demandé de conserver l'apparence générale de l'application et son fonctionnement. Au cours du stage, le contexte a profondément changé et nous avons dû nous réadapter aux nouvelles conditions.

En effet, nous avons dans un premier temps été confortés dans l'application du RGAA 1.0 par la parution du décret d'application de l'article 47 de la loi de 2005, le 14 mai 2009. Peu de temps après, nous avons vu paraître successivement le RGAA 2.0 le 3 juin 2009, puis le RGAA 2.1 le 3 juillet 2009 et pour finir le RGAA 2.2 le 30 septembre 2009.

Notre objectif s'est alors réadapté afin de mettre en conformité X-TEK par rapport au RGAA 2.2. Nous essayerons aussi de conserver l'apparence de l'application et son fonctionnement.

## 1.3. Présentation de X-TEK

X-TEK est un générateur de plateformes collaboratives qui s'apparente à un système de gestion de contenu (CMS<sup>6</sup> en anglais). Il répond aux besoins exprimés par l'IUT2 à la fin des années 90, en termes de rationalisation et d'optimisation de son système d'information, de développement du travail collaboratif et de simplification des communications autant internes qu'externes. Nous allons tout d'abord présenter les CMS de manière générale avant d'aborder X-TEK.

### 1.3.1. Les CMS

Les CMS sont des applications Web permettant la gestion de contenus (généralement du texte, des images et des vidéos). Ils offrent habituellement des fonctionnalités de travail collaboratif, de séparation du contenu de la mise en forme et le stockage du contenu en base de données.

Il existe plusieurs types de CMS, certains généralistes comme Drupal ou Joomla permettant de générer pratiquement toutes sortes de solutions et d'autres plus orientés comme :

- Wordpress ou Textpattern pour les blogs.
- osCommerce ou PrestaShop pour les sites de e-commerce.
- Moodle ou eFront pour l'enseignement à distance.

D'un point de vue architecture, un CMS est habituellement composé d'un serveur Web, d'un moteur de script et d'un système de gestion de bases de données. Les CMS possèdent, entre autres, un moteur de templates<sup>7</sup> permettant de séparer la présentation de la logique applicative et du contenu. Le moteur génère les pages Web en utilisant des templates écrits en HTML et possédant des mots clés qu'il substitue par les données récupérées lors de l'exécution de

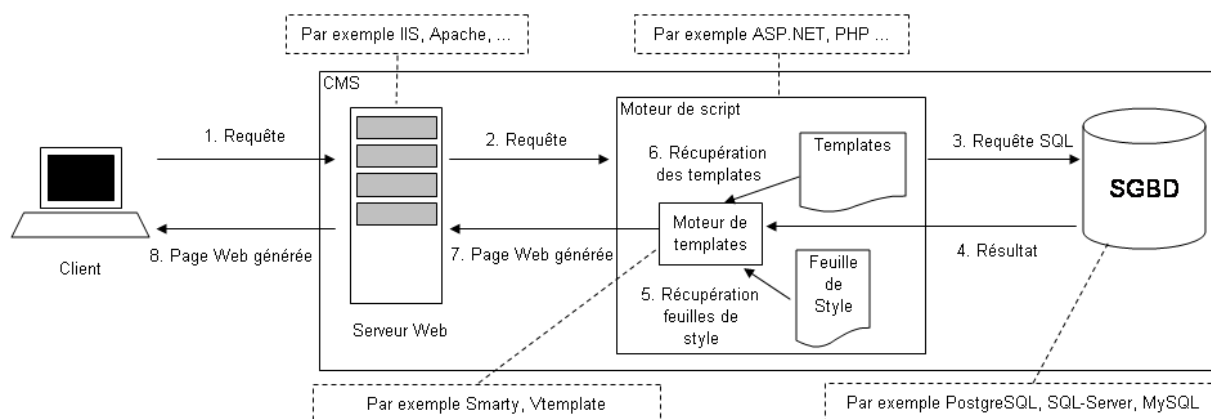
---

<sup>6</sup> CMS : Content Management System

<sup>7</sup> Template : Anglicisme utilisé en informatique pour désigner un modèle de présentation des données ou de conception de logiciel. Nous pouvons le traduire par gabarit ou patron de mise en page.



scripts. Ces données proviennent généralement du SGBD<sup>8</sup> du CMS. La Figure 1 montre l'architecture générale d'un CMS.



**Figure 1 : Architecture générale d'un CMS**

### 1.3.2. X-TEK

X-TEK est un générateur de plateformes collaboratives qui permet de générer principalement trois types de plateformes, à l'aide de ses différents modules que sont :

- INTRATEK : permettant de créer des intranets, des extranets et des groupes de travail collaboratifs.
- DIDACTEK : permettant de créer des plateformes d'enseignement à distance.
- WEBTEK : permettant de créer des sites Web avec gestion collaborative des contenus.

Il est composé d'un noyau principal représentant 80% des fonctionnalités communes à l'ensemble des modules Intratek, Didactek et Webtek. Les 20% restants sont constitués des fonctionnalités spécifiques de chacun des modules.

Toute plateforme générée partagera ainsi 80% de son code source avec les autres plateformes. L'avantage de cette mise en commun du code source est de faciliter l'application des mises à jour des plateformes ainsi que les correctifs.

<sup>8</sup> SGBD : Système de Gestion de Bases de Données

Il y a quelques années, la génération de plateformes était le principal avantage X-TEK. De nos jours, la génération de plateforme est implémentée dans d'autres CMS comme par exemple dans Joomla grâce à son extension JMS (Joomla Multi Site) et dans une moindre mesure dans Drupal.

### 1) Architecture

X-TEK possède la même architecture générale que les autres CMS. Techniquement, il s'appuie sur l'environnement LAMP<sup>9</sup> :

- Système Linux en distribution Debian ;
- Serveur Web Apache 2 ;
- Système de gestion de base de données MySQL 5 ;
- Moteur de script PHP<sup>10</sup> 5.

Il utilise les technologies suivantes :

- Le moteur de templates Smarty ;
- Le langage Javascript pour les scripts côté client ;
- Le langage XHTML<sup>11</sup> Transitional pour coder les pages Web qu'il génère ;
- Le langage CSS<sup>12</sup> 2.1 pour ses feuilles de style.

### 2) La gestion des contenus

Au sein de X-TEK, la gestion des contenus est relativement simple puisqu'elle s'articule essentiellement autour de pages auxquelles nous pouvons lier tout type de document quel que soit leur format de fichier. Les pages sont alors rangées dans des dossiers selon trois niveaux d'arborescence. Les différents niveaux sont les suivants :

---

<sup>9</sup> LAMP : Linux Apache MySQL PHP

<sup>10</sup> PHP : Hypertext Preprocessor

<sup>11</sup> XHTML : Extensible Hypertext Language est un successeur de l'HTML basé sur l'XML.

<sup>12</sup> CSS : Cascading Style Sheets est un langage de présentation des documents HTML.

- Catégorie : pouvant contenir des rubriques ou simplement une page.
- Rubrique : pouvant contenir des sous-rubriques ou simplement une page.
- Sous-rubrique : pouvant contenir uniquement des pages.

Il est à noter qu'il existe aussi des intercalaires qui ne sont pas considérés au niveau de X-TEK comme un véritable niveau d'arborescence mais plus comme un conteneur de catégories.

X-TEK offre différents modules applicatifs ou de mise en page communs aux différentes plateformes tels qu'une messagerie, un forum, un lecteur de flux RSS<sup>13</sup> et bien d'autres. Cependant le principal module, constituant la brique de base de X-TEK, est le module « dossier une seule page » permettant de créer des pages simples. La création d'une page est d'ailleurs facilitée par l'usage d'un éditeur de texte enrichi de type WYSIWYG<sup>14</sup>.

En plus de son éditeur de texte enrichi, X-TEK possède un mécanisme de masque de publication permettant la création de pages et la mise en ligne de tout type de document et de ressource. Les masques peuvent être créés à volonté et associés à différents modules permettant de gérer des pages simples.

### **1.4. Plan du Mémoire**

Nous commencerons, dans une première partie, par un bref rappel de l'accessibilité du Web à travers les deux versions des normes d'accessibilité et les deux versions du RGAA qui en découlent.

Puis, nous présenterons notre audit de l'accessibilité de X-TEK. Au terme de ce chapitre, nous donnerons les recommandations que nous avons exposées aux membres du projet X-TEK, ainsi que le plan de mise en conformité retenu suite à cette réunion.

---

<sup>13</sup> RSS : « Rich Site Summary » ou « Really Simple Syndication » est un langage XML utilisé pour la syndication de contenu Web.

<sup>14</sup> WYSIWYG : « What You See Is What You Get » (ce que vous voyez est ce que vous obtenez)

Au chapitre suivant, nous traiterons de la mise en conformité de X-TEK en commençant par une brève présentation de fonctionnement interne de l'application ; puis en passant en revue les différentes thématiques du RGAA 2.2 traitées lors de la mise en conformité du module « dossier une seule page ».

Enfin, nous dresserons le bilan de notre travail ainsi que les développements futurs envisageables, pour terminer par mon bilan personnel de cette année de stage au sein du service informatique de l'IUT2.



## 2. Rappel sur l'accessibilité du Web

Nous allons, dans un premier temps, présenter les normes de l'accessibilité créées par le W3C puis nous résumerons les deux référentiels d'accessibilité créés par la DGME et s'appuyant sur ces normes.

### 2.1. Les normes

Le W3C s'est penché sur les problèmes d'accessibilité du Web dans le milieu des années 90. Un groupe de travail, le WAI<sup>15</sup>, a été créé le 7 avril 1997 afin d'apporter des solutions à ces problèmes. Le WAI a permis la création d'une première norme, le WCAG 1.0 [WCAG10], devenu recommandation du W3C le 5 mai 1999 qui a servi de référence à toutes les initiatives d'accessibilité du Web jusqu'à nos jours. Elle proposait des solutions concrètes aux problèmes d'accessibilité.

Depuis 1999, les technologies du Web ont sensiblement évolué et le WCAG 1.0 apparaît un peu inadapté aux actuelles exigences des services internet. En effet, la norme traitait des technologies dont certaines sont devenues obsolètes et ne laissait pas de place aux nouvelles technologies. Une nouvelle version du WCAG est parue le 11 décembre 2008. Le WCAG 2.0 [WCAG20] se détache de l'application concrète à des technologies existantes et tente de rester le plus générique possible, afin de pouvoir s'adapter aux technologies futures. Le W3C offre tout de même, dans des documents annexes, des techniques permettant d'appliquer concrètement le WCAG 2.0 [UWCAG2] [TWCAG2] aux problèmes techniques actuels.

Nous donnons, dans ce qui suit, un aperçu de ces deux normes.

---

<sup>15</sup> WAI : Web Accessibility Initiative. Il s'agit du groupe de travail du W3C travaillant sur l'accessibilité du Web.

### **2.1.1. WCAG 1.0**

Cette norme définit 14 directives composées de plusieurs points de contrôle auxquels est rattachée une priorité. La liste des points de contrôle est donnée en annexe (Cf. annexe 1). Il existe trois niveaux de priorité numérotés de 1 à 3 pour ces points de contrôle :

- Le premier niveau est fondamental pour l'accessibilité. S'il n'est pas atteint, un ou plusieurs groupes de handicaps seraient dans l'incapacité d'accéder à l'information du site Web.
- Le deuxième niveau est important pour l'accessibilité et permet de garantir un accès pour tous à l'information.
- Le troisième niveau améliore sensiblement le confort d'accessibilité.

En validant les différents points de contrôle, un niveau de conformité est donné au site selon trois niveaux :

- le niveau « A » est atteint lorsque tous les points de contrôle de niveau « 1 » sont satisfaits.
- Le niveau « Double-A » est atteint lorsque tous les points de contrôle de niveau « 1 » et « 2 » sont satisfaits.
- Le niveau « Triple-A » est atteint lorsque tous les points de contrôle sont satisfaits.

Les différentes directives du WCAG 1.0 sont les suivantes :

#### **1) Fournir des alternatives équivalentes au contenu auditif et visuel**

Elle vise à rendre accessible les images en leur donnant des alternatives textuelles, ainsi que les contenus multimédias avec différents moyens tels que des sous-titrages, des descriptions de contenus visuels ainsi que des audios descriptions.

### **2) Ne pas s'en remettre uniquement aux couleurs**

Cette directive vise à rendre l'information parfaitement compréhensible pour les utilisateurs ayant des problèmes avec les couleurs comme dans le daltonisme et même pour les malvoyants. Elle permet aussi de rendre l'affichage compatible avec les afficheurs monochromes.

### **3) Utiliser le balisage et les feuilles de style, et cela, de façon appropriée**

Cette directive vise à faire respecter l'usage des balises et des feuilles de style qui sont souvent détournées pour créer des effets de présentation, ce qui restreint l'accessibilité.

### **4) Clarifier l'utilisation du langage naturel**

Cette directive vise à permettre l'identification de la langue utilisée dans la page, aussi bien la langue principale, que les passages dans une autre langue. Il est demandé aussi de définir toutes les abréviations et tous les acronymes en utilisant le balisage adéquat.

### **5) Créer des tableaux qui se transforment de façon élégante**

Cette directive vise à permettre l'usage convenable des tableaux, afin de faciliter leur linéarisation ou la navigation entre les cellules lorsque les logiciels d'aide le permettent.

### **6) S'assurer que les pages qui contiennent de nouvelles technologies se transforment de façon élégante**

Cette directive vise à faire en sorte que des pages créées avec les dernières technologies soient tout de même accessibles aux utilisateurs de navigateurs ne les supportant pas ou bien lorsque ces dites technologies sont désactivées sur les navigateurs utilisés.

### **7) Assurer à l'utilisateur le contrôle des changements du contenu lorsque ce dernier varie dans le temps**

Cette directive vise à donner aux utilisateurs le contrôle des éléments en mouvement et à éviter de mettre à jour automatiquement le contenu des pages sans que l'utilisateur puisse bloquer ces mécanismes.



**8) Assurer un accès direct aux interfaces utilisateur intégrées**

Cette directive vise à ce que, lorsqu'un objet inclus dans la page possède sa propre interface, cette interface soit accessible. Si on ne peut rendre accessible l'interface de l'objet, il faut proposer une solution alternative.

**9) Concevoir en respectant l'indépendance par rapport au périphérique**

Cette directive vise à garantir une navigabilité des pages en utilisant les périphériques d'entrée/sortie de prédilection de l'utilisateur. Il arrive souvent que certaines parties d'un site se limitent à une interaction à l'aide d'un périphérique de pointage comme la souris. En somme, cette directive demande que l'on puisse naviguer entièrement au clavier.

**10) Utiliser des solutions intermédiaires**

Cette directive vise à prendre en compte les lacunes et défauts des anciens logiciels d'assistance et navigateurs pour que les pages soient accessibles. Cependant, les points de cette directive sont obligatoires uniquement à la date de parution de la norme tant que ces problèmes n'ont pas été résolus. De nos jours, les logiciels de revue d'écran interprètent convenablement ces points.

**11) Utiliser des technologies et directives W3C**

Cette directive vise à promouvoir l'utilisation de technologies issues du W3C car elles reposent sur une conception ouverte et des normes. Il faut cependant respecter les directives d'accessibilité. Elle conseille d'éviter les technologies non W3C et non standard pour produire des pages plus accessibles mais, dans le cas où cela ne serait pas possible, elle recommande la création des pages alternatives.

**12) Fournir des informations de contexte et d'orientation**

Cette directive vise à clarifier les liens entre les différents éléments constituant la page afin de faciliter la compréhension des relations complexes entre ces éléments.

### **13) Fournir des mécanismes de navigation clairs**

Cette directive vise à faciliter la navigation sur le site Web en mettant en place des solutions de navigation compréhensibles comme des informations d'orientation, des barres de navigation ou bien une carte du site. Ceci facilite la navigation sur le site pour les malvoyants et la personne atteinte de problèmes cognitifs.

### **14) S'assurer que les documents soient clairs et simples**

Cette directive vise à privilégier la simplicité du contenu pour maximiser sa compréhension.

Il faut donc utiliser le langage le plus clair et le plus simple possible adapté au contenu du site et créer un style de présentation cohérent pour toutes les pages. Si cela peut faciliter la compréhension d'une page, on peut aussi associer du contenu visuel et audio au texte.

## **2.1.2. WCAG 2.0**

La norme WCAG 2.0 reprend le WCAG 1.0 et le réadapte aux contraintes modernes en s'affranchissant des technologies employées. Ceci permet de rendre cette nouvelle norme applicable à toute technologie actuelle ou future.

Elle est divisée en quatre principes d'accessibilité qui sont eux-mêmes divisés en plusieurs directives. Les directives ne correspondent plus aux directives du WCAG 1.0. En effet, leur découpage n'est plus le même et elles ne sont plus divisées en points de contrôle. A la place, elles comportent des critères de succès testables et répartis en trois niveaux numérotés « A », « AA » et « AAA ». La liste des critères de succès est donnée en annexe (Cf. annexe 2).

Nous retrouvons donc trois niveaux de conformité pour les pages Web :

- Niveau A : lorsque tous les critères de succès de niveau « A » sont respectés ou bien qu'une version alternative conforme est fournie.
- Niveau AA : lorsque tous les critères de succès de niveau « A » et « AA » sont respectés ou bien qu'une version alternative conforme au niveau « AA » est fournie.

- Niveau AAA : lorsque tous les critères de succès de niveau « A », « AA » et « AAA » sont respectés ou bien qu'une version alternative conforme au niveau « AAA » est fournie.

Les quatre principes d'accessibilité sont les suivants :

**1) Perceptible : L'information et les composants d'interface utilisateur doivent être présentés à l'utilisateur de façon à ce qu'il puisse les percevoir**

Ce principe d'accessibilité regroupe toutes les notions liées à la perception des pages Web et les regroupe dans quatre directives :

- La première directive traite des alternatives textuelles aux contenus non textuels. De plus, cette directive prend en compte des situations qui étaient peu fréquentes en 1999 comme les tests de Turing, tests souvent employés afin d'empêcher des robots de créer automatiquement des comptes de messagerie électronique et par la suite d'émettre des SPAMS<sup>16</sup>.
- La deuxième directive traite des alternatives aux médias basés sur le temps, c'est-à-dire les contenus vidéo et audio préenregistrés.
- La troisième directive traite de la création de contenus adaptables qui puissent être présentés de différentes manières sans perte d'information ou de structure.
- La quatrième directive traite de la création de contenus distinguables afin de faciliter, aux utilisateurs, la lecture ou l'écoute du contenu.

---

<sup>16</sup> SPAM : pourriel ou pollurriel est une communication électronique non sollicitée

**2) Utilisable : Les composants de l'interface utilisateur et de navigation doivent être utilisables**

Ce principe d'accessibilité regroupe les points à respecter afin qu'un site soit manipulable par tout le monde. Pour ce faire, il définit quatre directives :

- La première directive impose que toutes les fonctionnalités du site soient accessibles au clavier.
- La deuxième directive impose de donner assez de temps à l'utilisateur afin qu'il puisse lire et utiliser le contenu.
- La troisième directive impose de ne pas créer de contenu susceptible de provoquer des crises d'épilepsie chez certaines personnes.
- La quatrième directive vise à fournir tous les moyens possibles pour faciliter la navigation sur le site et accéder rapidement au contenu désiré.

**3) Compréhensible : Les informations et l'utilisation de l'interface utilisateur doivent être compréhensibles**

Ce principe d'accessibilité regroupe les points à respecter afin de rendre un site compréhensible et logique. Il définit trois directives :

- La première directive traite du contenu des pages et vise à les rendre lisibles et compréhensibles.
- La deuxième directive traite de la prédictibilité des pages Web afin de donner une structure logique aussi bien interne à la page que dans ses relations avec d'autres pages.
- La troisième directive traite des aides à apporter à l'utilisateur dans son interaction avec la page, aussi bien au niveau de la saisie de contenu qu'au niveau de l'identification du contenu.

- 4) **Robuste : Le contenu doit être suffisamment robuste pour être interprété de manière fiable par une large variété d'agents utilisateurs, y compris les technologies d'assistance**

Ce principe d'accessibilité regroupe les points à respecter, afin de rendre les pages compatibles avec les technologies actuelles et les futures technologies. Il est défini une seule directive qui traite de ce sujet.

## **2.2. Les référentiels d'accessibilité de la DGME**

La DGME s'est appuyée sur le WCAG 1.0 et la méthode unifiée UWEM<sup>17</sup> afin de créer son premier référentiel d'accessibilité le RGAA 1.0. Il aura fallu plus de 3 ans entre la promulgation de la loi de 2005 et la finalisation du RGAA 1.0 le 24 novembre 2008.

L'adéquation du RGAA 1.0 avec les normes d'accessibilité fut de courte durée puisque le WCAG 2.0 est devenu une recommandation du W3C moins d'un mois après la parution du RGAA 1.0.

La DGME s'est alors attelée à la conception d'un nouveau référentiel. Le RGAA 2.0 est paru le 3 juin 2009 suivi de près par deux autres versions le RGAA 2.1 et 2.2, parues respectivement le 3 juillet et le 30 septembre 2009.

Nous présentons rapidement les caractéristiques de ces deux référentiels.

### **2.2.1. RGAA 1.0**

Le RGAA 1.0 [RGAA10] s'appuie entièrement sur le WCAG 1.0 en reprenant l'ensemble des directives et points de contrôle le constituant. Chaque point de contrôle est alors subdivisé en plusieurs tests permettant de vérifier la conformité d'une page Web. Il fournit un cadre d'application de la norme et définit des obligations de réussite sur trois ans pour les différents tests.

---

<sup>17</sup> UWEM : UWEM, ou Unified Web Evaluation Methodology, est une méthode d'évaluation de l'accessibilité du Web résultant du travail conjoint de trois projets européens (EIAO, Support EAM et BenToWeb)

### **2.2.2. RGAA 2.2**

Le RGAA 2.2 [RGAA22] s'appuie entièrement sur le WCAG 2.0 et reprend la logique des tests unitaires. Il s'articule autour de cinq documents permettant de séparer la logique du référentiel. Ainsi nous avons :

- un document expliquant le fonctionnement du référentiel ;
- un guide d'accompagnement ;
- une annexe faisant le lien entre les critères de chaque directive du WCAG 2.0 et les tests unitaires du RGAA 2.2 ;
- une annexe contenant l'ensemble des tests unitaires du RGAA 2.2 ;
- enfin une annexe mettant en forme, dans un tableau, les liens entre les principes d'accessibilité, les directives et les critères du WCAG 2.0, et les tests unitaires du RGAA 2.2.

Chaque test unitaire permet de vérifier la conformité d'une page vis-à-vis d'un critère et fournit les différentes techniques afférentes disponibles dans les documents annexes du WCAG 2.0.

## **2.3. Synthèse**

Nous avons présenté rapidement, dans ce chapitre, les deux normes internationales régissant l'accessibilité du Web. La dernière en date, le WCAG 2.0 paru en décembre 2008, remplace le WCAG 1.0 paru en mai 1999. Nous avons aussi présenté les deux référentiels produits par la DGME, le RGAA 1.0 et le RGAA 2.2. Étant devenu obsolète seulement un mois après sa parution, le RGAA 1.0 a été rapidement remplacé par le RGAA 2.0 en juin 2009. Cette dernière version est conforme au WCAG 2.0 et offre une application concrète de la norme à travers des tests unitaires.

Dans le chapitre suivant, nous aborderons l'audit d'accessibilité de l'outil X-TEK, étude préalable afin de définir un plan de mise en conformité.



### 3. Audit de l'accessibilité de X-TEK

L'audit s'étant déroulé au début de stage et donc avant la parution du RGAA 2.0, il repose entièrement sur des tests du WCAG 1.0 et RGAA 1.0. Bien que les résultats ne correspondent pas entièrement avec les nouveaux tests WCAG 2.0 et RGAA 2.0, ils restent suffisamment révélateurs du niveau d'accessibilité générale de X-TEK avant sa mise en conformité.

L'objectif était d'évaluer le niveau d'accessibilité générale de l'outil X-TEK par rapport au RGAA<sup>18</sup> et de définir un plan de mise en conformité de X-TEK. Comme la phase de mise en conformité réelle a suivi l'évolution du référentiel et est donc basée sur le RGAA 2.0, tout rapprochement avec le RGAA 1.0 est devenu inutile. Nous nous contenterons donc de la seule partie WCAG 1.0 des tests et nous ferons l'impasse sur l'étude RGAA 1.0 qui fut effectuée lors de cette phase d'audit qui, pour rappel, portait sur la priorisation des réussites aux tests RGAA 1.0 sur trois ans.

Pour cette étude, il n'est pas question de tester une plateforme en exploitation mais de tester la capacité de l'outil à générer des pages accessibles, hors contenu généré par l'utilisateur. En effet, il aurait été peu judicieux de procéder autrement car les erreurs génériques auraient été noyées dans les erreurs de contenu.

En premier lieu, nous avons mis en place une plateforme de test permettant d'exploiter l'ensemble des possibilités de l'application. Puis nous nous sommes mis en quête de logiciels permettant d'automatiser les tests et générer des rapports d'accessibilité page par page. Dans un deuxième temps, nous sommes passés à la phase de test à part entière et avons rassemblé l'ensemble des données recueillies pour les traiter.

Après une analyse des résultats, nous avons défini des recommandations quant à la démarche pour mettre X-TEK en conformité. Nous avons exposé ces recommandations au cours d'une réunion de définition du plan de mise en conformité d'X-TEK.

---

<sup>18</sup> RGAA : Référentiel Général d'Accessibilité pour les Administrations



## **3.1. Mise en place d'une plateforme de test**

Notre plateforme de test est basée sur la plateforme WEBTEK-98. Nous avons choisi cette plateforme pour deux raisons :

- Elle est la plus avancée en terme de charte graphique.
- Les WEBTEK sont les plateformes les plus tournées vers le public et donc les plus sujettes au respect des normes d'accessibilité.

Dans notre étude d'accessibilité, nous avons testé l'ensemble des modules disponibles, ainsi que les différents styles de présentation des rubriques et sous-rubriques offerts par X-TEK. Dans ce qui suit, nous présentons les différents modules disponibles, la structure de la plateforme de test et les différentes pages Web utilisées pour tester l'application.

### **3.1.1. Les différents modules disponibles**

#### **1) Dossier une seule page**

Il s'agit de la brique de base de X-TEK permettant de créer une page. La création est effectuée en utilisant l'éditeur de texte enrichi FCKEditor (en version 2.6.3), fourni par X-TEK. La Figure 2 montre un exemple de formulaire d'édition d'un « dossier une seule page ». Cet éditeur de texte permet aussi de créer une page web en exploitant directement les fonctions de glissé-déposé permettant, de ce fait, de copier le contenu d'une page existante sur le Web. De plus, il permet aussi de saisir directement du code HTML<sup>19</sup> en passant dans le mode HTML.

---

<sup>19</sup> HTML : Hypertext Markup Language est le langage conçu pour représenter les pages Web

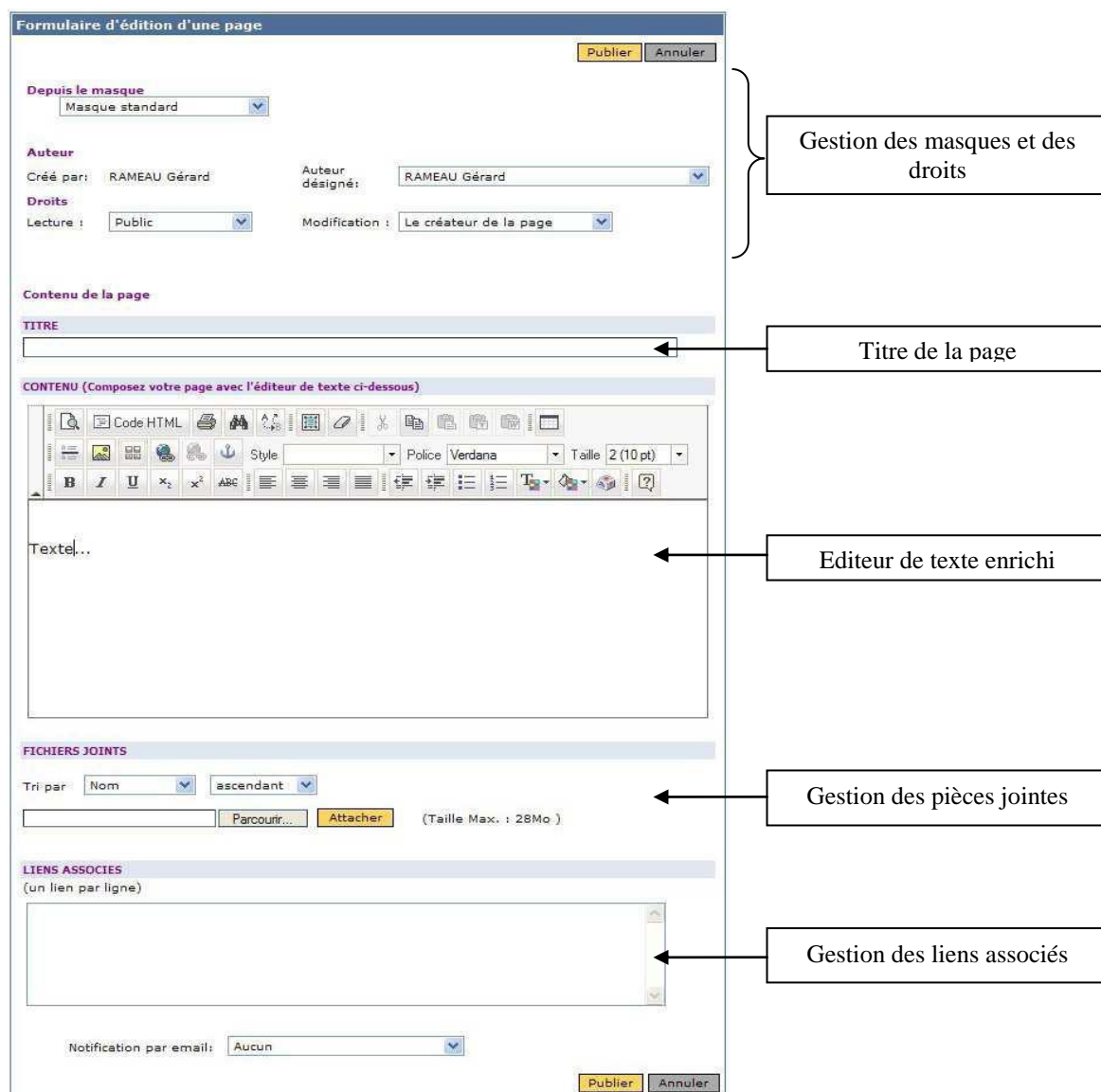


Figure 2 : Formulaire d'édition d'une page

## 2) Dossier liste

Un « dossier liste » permet de présenter, sous forme de liste, un ensemble de « dossier une seule page » et pouvant contenir plusieurs niveaux de sous-dossier que l'on nomme chapitre. La Figure 3 montre un exemple de « dossier liste » comportant des pages et des chapitres. La consultation de la page offre un menu de navigation permettant de passer d'une page à l'autre dans la liste, de revenir à la première page ou d'aller à la dernière, ainsi que de revenir à la liste.



Figure 3 : Dossier liste

### 3) Dossier menu

Un « dossier menu » permet de présenter un maximum de huit « dossier une seule page » dans un menu. Comme le montre la Figure 4, le menu se situe en partie haute du contenu de la page et le « dossier une seule page » courant est affiché en dessous avec, selon les cas, le titre de la page.

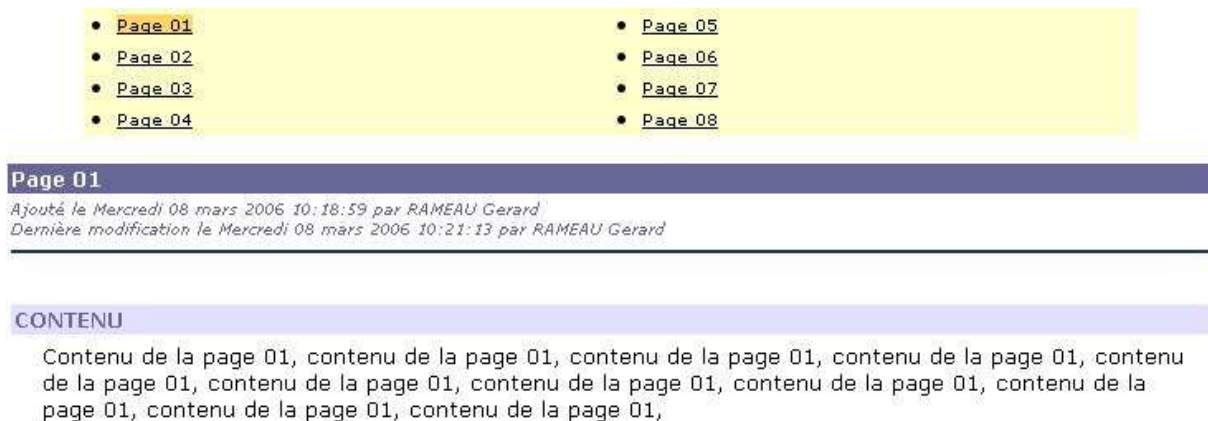


Figure 4 : Dossier menu

#### **4) Page Web externe**

Ce module permet d'inclure dans X-TEK une page provenant d'un autre site. Elle sera insérée dans une page à l'aide d'une « iframe<sup>20</sup> ».

#### **5) Calendrier**

Le calendrier repose sur « webcalendar » version 0.9.43. Il offre les fonctionnalités standards d'un calendrier en permettant de créer des événements, en les visualisant jour par jour, par semaine, par mois, par année et en liste.

#### **6) Forum**

Le forum repose sur l'application « phorum » en version inconnue. Ce module offre les fonctions de base des forums ainsi qu'une fonction de recherche dans les forums.

#### **7) Messagerie interne**

C'est une application de messagerie standard développée pour X-TEK. Elle est utilisable uniquement par les utilisateurs connectés.

#### **8) Trombinoscope**

C'est une application permettant d'afficher les utilisateurs sous forme de liste ou de planche avec photo.

#### **9) Liste des utilisateurs**

C'est une application qui permet de lister les utilisateurs, en spécifiant différents champs d'information désirés et en effectuant des tris et des regroupements.

#### **10) Formulaire en ligne**

Le formulaire en ligne est un module permettant de créer des formulaires. Il s'appuie sur le système de masques de X-TEK. Afin de créer un formulaire, il faut au préalable créer le

---

<sup>20</sup> iframe : élément HTML permettant d'incorporer des cadres

masque correspondant dans la partie administration puis l'associer avec le formulaire lors de sa création.

## **11) Player RSS**

Le player RSS permet de récupérer des flux RSS dont nous avons spécifié l'url.

### **3.1.2. La structure de la plateforme de test**

Afin de permettre l'utilisation de l'ensemble des possibilités offertes par X-TEK, nous avons généré une structure de pages faisant appel à l'ensemble des modules, en utilisant les différentes présentations des rubriques et sous-rubriques, ainsi que toutes les possibilités offertes par l'éditeur de texte enrichi.

Pour ce faire, nous avons complété notre plateforme avec un intercalaire de test, nommé « TEST FRONT ». Cet intercalaire contient quatre catégories dans lesquelles nous mettons en œuvre les différentes facettes de l'outil X-TEK :

- Page menu onglets
- Page menu hypermots
- Page menu déroulant
- Les modules

Le contenu de ces quatre catégories est exposé dans la suite. Afin d'effectuer nos tests d'accessibilité, nous avons sauvegardé certaines pages représentatives générées par l'application à partir de cette structure et des pages issues de la partie administrative de l'application (c'est-à-dire les fonctionnalités d'administration des dossiers et des pages).

#### **1) Catégorie « Page menu onglets »**

Nous avons défini cette catégorie, afin de mettre en œuvre l'affichage des rubriques sous forme d'onglets, en imbriquant des sous-rubriques présentées en Hypermot et menu déroulant. De plus, nous mettons en œuvre, plusieurs fois dans cette catégorie, les modules « Dossier

une seule page » et « Dossier liste ». L'arborescence de cette catégorie est présentée dans la Figure 5.

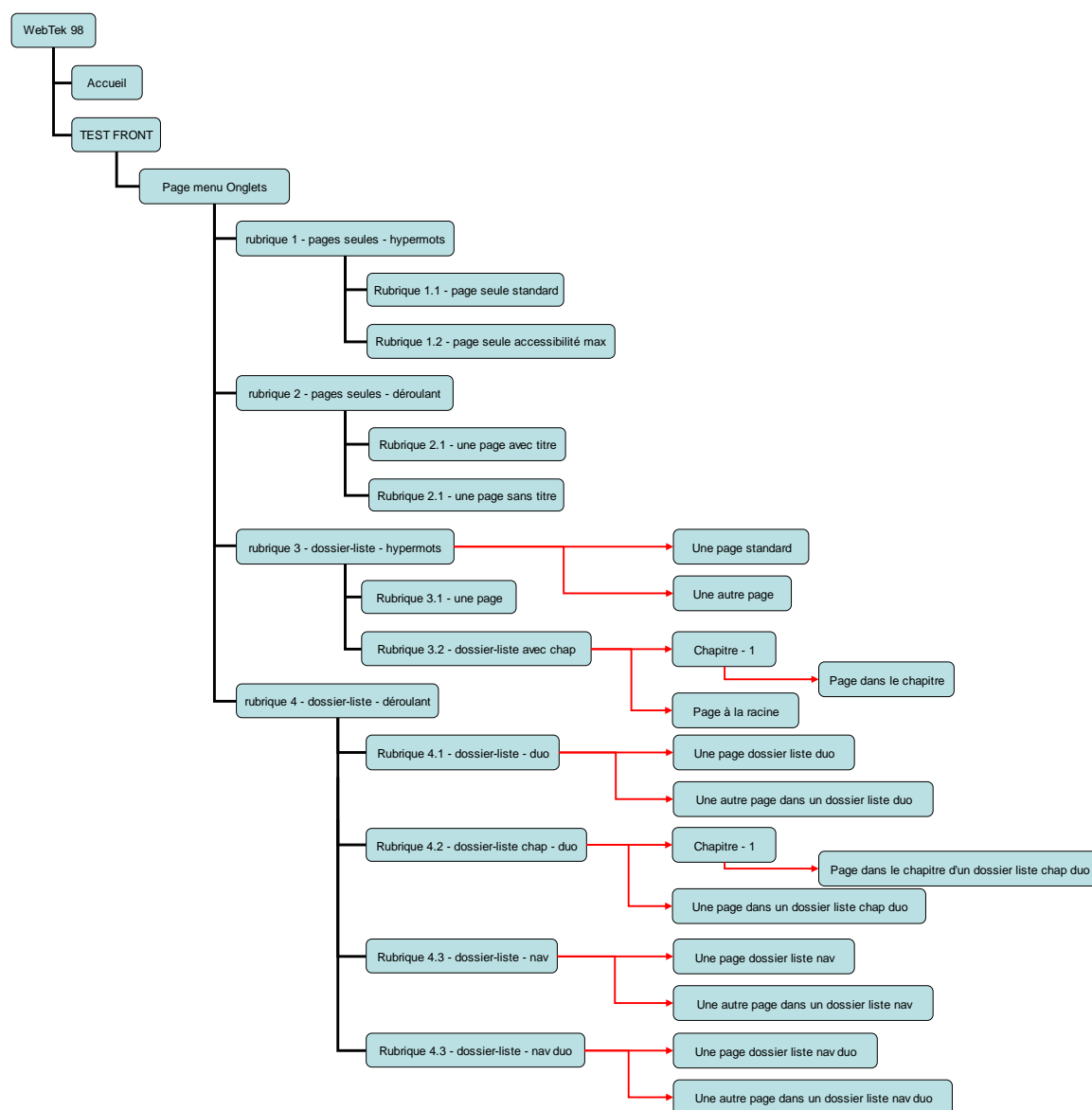


Figure 5 : Arborescence de la catégorie « Page menu onglets »

A partir de cette catégorie, nous enregistrons plusieurs pages web permettant de tester les différentes fonctionnalités des modules testés conformément au Tableau 1. Il est à noter qu'au niveau des pages « TEST FRONT/Page menu onglet/rubrique 1.1 - page seule standard » et « TEST FRONT/Page menu onglet/rubrique 1.2 - page seule accessibilité max » nous mettons en œuvre l'ensemble des possibilités de présentation disponibles au sein de l'éditeur de texte enrichi de la plateforme X-TEK. Les différences notables entre ces pages sont l'utilisation dans la page d'accessibilité maximale des champs de commentaires, de légende et de texte

alternatif offerts par l'éditeur de texte enrichi, afin de vérifier la capacité de l'éditeur de texte enrichi à générer des pages accessibles.

Description	Pages
La page d'accueil	Accueil
Différentes variantes de présentation de rubrique avec une seule page (module dossier une seule page)	TEST FRONT/Page menu onglets
	TEST FRONT/Page menu onglets/rubrique 1 - pages seules - hypermots
	TEST FRONT/Page menu onglets/rubrique 1.1 - page seule standard
	TEST FRONT/Page menu onglets/rubrique 1.2 - page seule accessibilité max
	TEST FRONT/Page menu onglets/rubrique 2 - pages seules - déroulant
	TEST FRONT/Page menu onglets/rubrique 2.1 - une page avec titre
	TEST FRONT/Page menu onglets/rubrique 2.2 - une page sans titre
Un dossier liste standard (module dossier liste)	TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste - hypermots
Test des fonctionnalités d'un dossier liste standard (module dossier liste)	TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste - hypermots (fonction Recherche)
	TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste - hypermots (fonction Visualisation du contenu de "Une autre page")
Test des fonctionnalités de modification, de copie et déplacement d'une page (module dossier une seule page)	TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste - hypermots (fonction Edition d'une page)
	TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste - hypermots (fonction Copie d'une page)
	TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste - hypermots (fonction Déplacement d'une page)
Un dossier liste avec chapitre (module dossier liste)	TEST FRONT/Page menu onglets/rubrique 3.2 - dossier-liste avec chap
Test des fonctionnalités d'un dossier liste avec chapitre différent des standards (module dossier liste)	TEST FRONT/Page menu onglets/rubrique 3.2 - dossier-liste avec chap (fonction Recherche)
	TEST FRONT/Page menu onglets/rubrique 3.2 - dossier-liste avec chap (fonction Visualisation du contenu du chapitre)
Une page d'un dossier liste navigation (module dossier liste)	TEST FRONT/Page menu onglets/rubrique 4.3 - dossier-liste - nav (fonction Visualisation d'une page)

**Tableau 1 : Liste des pages testées issues de la page d'accueil et de la catégorie « Page menu onglets »**

## 2) Catégorie « Page menu hypermots »

Nous avons défini cette catégorie afin de mettre en œuvre l'affichage des rubriques sous forme d'hypermots en imbriquant des sous-rubriques présentées en Hypermot et menu déroulant. De plus, nous mettons en œuvre dans cette catégorie, les modules « Dossier une seule page », « Dossier menu » et « Pages webs externes ». L'arborescence de cette catégorie est présentée dans la Figure 6.

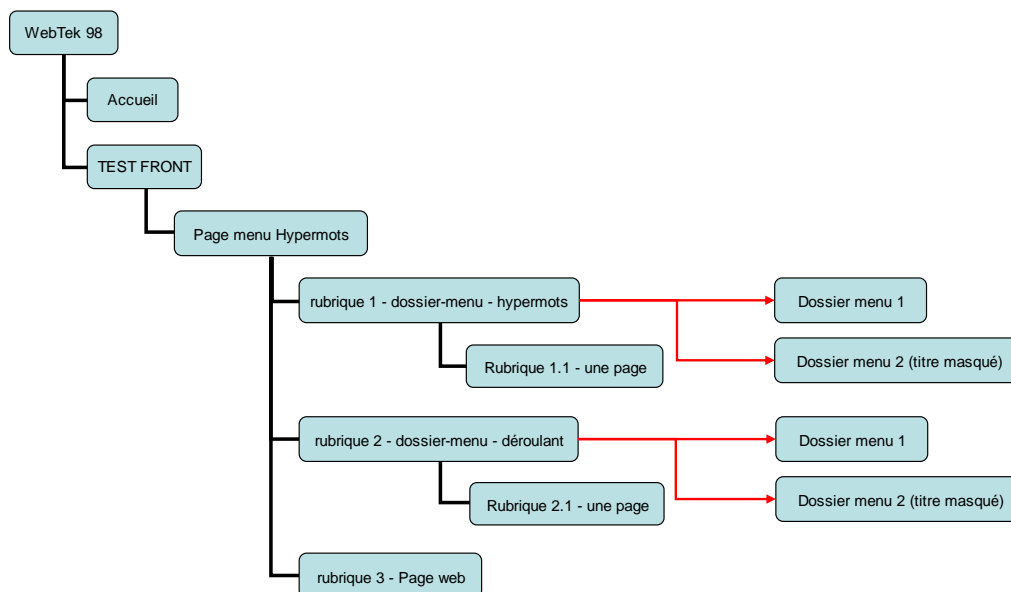


Figure 6 : Arborescence de la catégorie « Page menu hypermots »

A partir de cette catégorie, nous enregistrons plusieurs pages web permettant de tester les différentes fonctionnalités des modules testés conformément au Tableau 2.

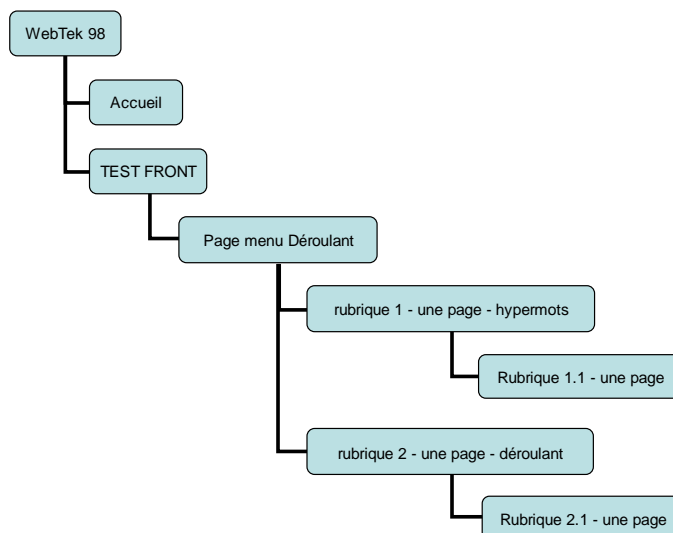
Description	Pages
Une rubrique de plus une seule page (dossier une seule page)	TEST FRONT/Page menu hypermots
Deux pages à l'intérieur d'un dossier menu avec ou sans titre masqué (module dossier menu)	TEST FRONT/Page menu hypermots/rubrique 1 - dossier-menu - hypermots (fonction Dossier menu 1)
	TEST FRONT/Page menu hypermots/rubrique 1 - dossier-menu - hypermots (fonction Dossier menu 2 (titre masqué))
Encore une page dossier menu (module dossier menu)	TEST FRONT/Page menu hypermots/rubrique 2 - dossier-menu - déroulant/Dossier menu 1
Un page web (module page web)	TEST FRONT/Page menu hypermots/rubrique 3 - Page web

Tableau 2 : Liste des pages testées issues de la catégorie « Page menu hypermots »

### 3) Catégorie « Page menu déroulant »

Nous avons défini cette catégorie, afin de mettre en œuvre l'affichage des rubriques sous forme de menu déroulant, en imbriquant des sous-rubriques présentées en Hypermot et menu déroulant. De plus, nous mettons en œuvre, dans cette sous-catégorie, les modules « Dossier une seule page ». L'arborescence de cette catégorie est présentée dans la Figure 7.





**Figure 7 : Arborescence de la catégorie « Page menu déroulant »**

A partir de cette catégorie, nous enregistrons plusieurs pages web permettant de tester les différentes fonctionnalités des modules testés conformément au Tableau 3.

Description	Pages
D'autres présentations de rubrique avec une seule page (module dossier une seule page)	TEST FRONT/Page menu déroulant
	TEST FRONT/Page menu déroulant/rubrique 1 - une page - hypermots
	TEST FRONT/Page menu déroulant/rubrique 2 - une page - déroulant

**Tableau 3 : Liste des pages testées issues de la catégorie « Page menu déroulant »**

#### 4) Catégorie « Les modules »

Nous avons défini cette catégorie, afin d'utiliser l'ensemble des modules particuliers. Nous mettons en œuvre, dans cette catégorie, les modules Calendrier, Forum, Messagerie, Trombinoscope, Liste des utilisateurs, Formulaire et Flux RSS. L'arborescence de cette catégorie est présentée dans la Figure 8.

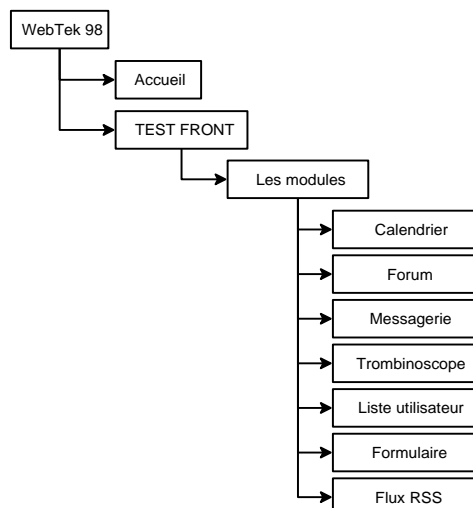


Figure 8 : Arborescence de la catégorie « Les modules »

A partir de cette catégorie, nous enregistrons plusieurs pages web permettant de tester les différentes fonctionnalités des modules testés conformément au Tableau 4.

Description	Pages
Test des fonctionnalités du calendrier (module calendrier)	TEST FRONT/Les modules/Calendrier (Affichage d'un mois)
	TEST FRONT/Les modules/Calendrier (Affichage d'un jour)
	TEST FRONT/Les modules/Calendrier (Affichage d'une semaine)
	TEST FRONT/Les modules/Calendrier (Affichage d'une année)
	TEST FRONT/Les modules/Calendrier (Affichage d'un événement)
	TEST FRONT/Les modules/Calendrier (Affichage de la liste d'événements)
	TEST FRONT/Les modules/Calendrier (Recherche dans la liste d'événements)
	TEST FRONT/Les modules/Calendrier (Ajout d'un événement)
Test des fonctionnalités du forum (module forum)	TEST FRONT/Les modules/Forum
	TEST FRONT/Les modules/Forum (Ajout d'un forum)
	TEST FRONT/Les modules/Forum (Affichage du forum 4)
	TEST FRONT/Les modules/Forum (Nouveau sujet dans le forum 4)
	TEST FRONT/Les modules/Forum (Recherche dans le forum 4)
	TEST FRONT/Les modules/Forum (Affichage du sujet de discussion 1 dans le forum 4)
	TEST FRONT/Les modules/Forum (Réponse au sujet de discussion 1 dans le forum 4)
	TEST FRONT/Les modules/Forum
Test des fonctionnalités de la messagerie (module messagerie)	TEST FRONT/Les modules/Messagerie (Affichage boîte de réception)
	TEST FRONT/Les modules/Messagerie (Affichage boîte d'envoi)
	TEST FRONT/Les modules/Messagerie (Nouveau message)
	TEST FRONT/Les modules/Messagerie (Lire un message)
	TEST FRONT/Les modules/Messagerie (Rechercher)
Test des fonctionnalités du trombinoscope (module trombinoscope)	TEST FRONT/Les modules/Trombinoscope
	TEST FRONT/Les modules/Trombinoscope (Affichage en planche)
	TEST FRONT/Les modules/Trombinoscope (Affichage d'un utilisateur)
Une liste utilisateur (module liste des utilisateurs)	TEST FRONT/Les modules/Liste utilisateur
Test des fonctionnalités du formulaire	TEST FRONT/Les modules/Formulaire
	TEST FRONT/Les modules/Formulaire (Consultation d'un formulaire)

Description	Pages
(module formulaire)	TEST FRONT/Les modules/Formulaire (Nouveau formulaire)
Une page de flux RSS (module flux RSS)	TEST FRONT/Les modules/Flux RSS

Tableau 4 : Liste des pages testées issues de la catégorie « Les modules »

## 5) Administration des dossiers

Nous avons aussi sauvegardé des pages Web issues de la partie administration de la plateforme. Nous testons ici les fonctionnalités de création, de modification, de déplacement et de copie de dossier conformément au Tableau 5.

Description	Pages
Test des fonctionnalités de création, modification, déplacement et copie de la partie administration d'X-TEK	Partie administration -- création d'une seule page
	Partie administration -- création d'un dossier liste
	Partie administration -- création d'un dossier liste avec chapitre
	Partie administration -- création d'un dossier menu
	Partie administration -- création d'une page web externe
	Partie administration -- création d'un calendrier
	Partie administration -- création d'un forum
	Partie administration -- création d'une messagerie
	Partie administration -- création d'un trombinoscope
	Partie administration -- création d'une liste utilisateur
	Partie administration -- création d'un formulaire
	Partie administration -- création d'un annuaire
	Partie administration -- création d'une page de flux rss
	Partie administration -- modification d'un dossier
Partie administration -- déplacement d'un dossier	
Partie administration -- copie d'un dossier	

Tableau 5 : Liste des pages testées issues de la partie administrative

## 3.2. La recherche de logiciels de tests d'accessibilité

Afin de permettre une analyse plus rapide de l'accessibilité de X-TEK, nous nous sommes mis en quête d'outils de tests automatisés et gratuits. Il existe plus d'une centaine d'applications de test, pour la plupart payantes, pour d'autres extrêmement limitées en termes de tests évalués et d'autres uniquement utilisables sur Internet. Le W3C nous propose une liste non exhaustive d'applications de test sur son site à l'adresse :

<http://www.w3.org/WAI/ER/tools/complete>

Pour faire un choix entre toutes ces applications, nous avons défini trois critères principaux :

1. L'application doit en priorité tester convenablement les pages selon les critères du WCAG 1.0
2. Elle doit pouvoir s'exécuter en local. Il est important de pouvoir tester notre plateforme de test qui est inaccessible sur le Web.
3. Elle doit générer un rapport d'évaluation énumérant les différentes erreurs et leur localisation dans le code source.

Après avoir essayé des dizaines d'applications, nous avons pu en isoler trois correspondant à nos critères :

- TAW<sup>21</sup> (version 3.08, développée par la fondation CTIC<sup>22</sup>)
- Total Validator (version 5.4)
- WAI<sup>23</sup> (version 5.11, développée par Fujitsu)

Chaque application signale des problèmes détectés automatiquement et nous avertit de possibles problèmes qu'il nous faut vérifier manuellement. TAW et WAI détectent beaucoup plus de problèmes d'accessibilité que Total Validator. Cependant, ce dernier permet de tester la conformité de l'HTML, ce que les deux autres ne font pas. Il est toutefois sujet à une explosion des problèmes de conformité HTML puisque, ne sachant pas s'abstraire des erreurs de structure précédentes, ce qui ne fait qu'invalider le code source suivant. Nous pouvons citer en exemple le cas le plus extrême que nous ayons constaté en testant le module « calendrier », pour lequel Total Validator nous a retourné 6260 erreurs dont 765 erreurs d'analyse syntaxique, 5085 erreurs d'HTML et 410 erreurs de WCAG 1.0 (6 de niveau A, 403 de niveau AA et 1 de niveau AAA).

---

<sup>21</sup> TAW : Test Accesibilidad Web

<sup>22</sup> CTIC : Centro Tecnológico de la Información y de la Comunicación. Il s'agit d'un centre de développement de l'information et des technologies des communications de la principauté des Asturies en Espagne. Cette organisation privée à but non lucratif est constituée par un groupe d'entreprises issues des nouvelles technologies et par le gouvernement de la principauté des Asturies.

<sup>23</sup> WAI : Web Accessibility Inspector

Ces applications ne sont malheureusement pas parfaites et leurs résultats diffèrent les uns des autres. Dans de telles conditions, nous ne sommes pas en mesure de savoir si les problèmes signalés sont réellement présents.

Nous avons dû faire un choix entre ces différentes applications afin d'isoler celle se rapprochant le plus du véritable niveau de conformité des pages testées. La démarche retenue pour faire notre choix est de prendre une page comportant le plus grand nombre d'erreurs différentes et de comparer les différents rapports d'accessibilité des applications avec un test manuel de la page. L'application offrant le rapport le plus proche sera retenue. Pour ce faire, nous avons donc choisi la page de création d'un formulaire à partir du menu d'administration et nous avons effectué le test manuel en nous reposant sur les tests du RGAA 1.0.

Deux points sont à préciser sur notre test manuel. Premièrement, les points de contrôle 10.3, 10.4 et 11.4 (Cf. annexe 1) étant considérés comme obsolètes au niveau du RGAA 1.0, ils n'ont pas été évalués dans le test manuel. Deuxièmement, au vu de notre maîtrise des différentes technologies et du référentiel au moment de ce test, nous avons rencontré des difficultés à appliquer sept des tests du RGAA 1.0, ce qui nous a obligé à signaler des erreurs probables non testées. Ces tests sont les suivants :

- 2.2.1 : « Valeur du ratio de contraste des caractères de petite taille contenus dans des éléments non textuels » générant 8 erreurs non testées.
- 2.2.4 : « Valeur du ratio de contraste des caractères de grande taille contenus dans des éléments non textuels » générant 1 erreur non testée.
- 6.3.3 : « Présence d'une alternative au code javascript » générant 1 erreur non testée.
- 6.5.3 : « Accessibilité des contenus dynamiques en javascript » générant 1 erreur non testée.
- 9.4.2 : « Absence de suppression javascript du focus clavier » générant 1 erreur non testée.
- 10.5.3 : « Maintien de la distinction visuelle des liens » générant 1 erreur non testée.

- 10.5.4 : « Absence de suppression visuelle d'espacement entre les liens » générant 1 erreur non testée.
- 12.3.1 : « Regroupement d'éléments de formulaire via l'élément fieldset » générant 1 erreur non testée.
- 13.5.1 : « Présence de menus ou de barres de navigation » générant 1 erreur non testée.

Les résultats sont donnés dans le Tableau 6. Afin de faire ressortir la correspondance des résultats des différentes applications, nous avons attribué une couleur de fond à chaque point de contrôle des différents rapports :

- en vert lorsque le résultat est proche du résultat manuel ;
- en orange lorsque le résultat correspond en partie ;
- en rouge lorsque le résultat n'est pas satisfaisant ;
- en gris lorsque le résultat trouve une correspondance du point de vue WCAG 1.0, mais n'apparaît pas dans les tests RGAA 1.0.

Chaque point de contrôle est évalué à l'aide de plusieurs tests, ce qui induit que le nombre d'erreurs pour un point de contrôle peut être composé de plusieurs erreurs différentes. C'est pourquoi, dans notre tableau, nous pouvons avoir des colorations différentes alors que nous avons pratiquement le même nombre d'erreurs ; ceci afin de faire ressortir la tendance réelle à l'intérieur du rapport. Si des problèmes signalés mais inexistantes sont majoritaires, alors le fond sera rouge ou bien gris s'ils n'ont pas de correspondance au niveau RGAA 1.0. De plus, si des erreurs sont signalées bien que nécessitant d'être testées manuellement, elles rentrent en compte dans la conformité du rapport avec le test manuel. Par exemple, au point de contrôle 1.1 (Cf. annexe 1), TAW signale 14 erreurs à vérifier alors que le test manuel en décèle réellement 11. Nous considérons toutefois que TAW est le plus proche puisque, dans les 14 erreurs à vérifier, les 11 repérées dans le test manuel sont mentionnées et il ne reste finalement que 3 erreurs à vérifier.

Audit de l'accessibilité de X-TEK

Point de contrôle	Test manuel						TAW						Validator						WAI					
	Testé			Non testé			Auto			Manuel			Auto			Manuel			Auto			Manuel		
	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA
1.1	11	0	0	0	0	0	0	0	0	14	0	0	5	0	0	0	0	0	6	0	0	0	0	0
1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.2	0	0	0	0	9	0	0	0	0	9	1	0	0	0	0	0	0	0	67	0	0	0	0	0
3.1	0	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.2	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.3	0	36	0	0	0	0	0	0	0	16	0	0	0	7	0	0	0	0	0	0	0	0	0	0
3.4	0	1	0	0	0	0	0	18	0	2	0	0	10	0	0	0	0	0	1	0	0	60	0	0
3.5	0	6	0	0	0	0	0	1	0	1	0	0	4	0	0	0	0	0	0	0	0	0	0	0
3.6	0	11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.7	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4.3	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
5.1	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.2	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.3	0	24	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.5	0	0	0	0	0	0	0	0	11	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
5.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.1	11	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.2	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.3	22	0	0	1	0	0	10	0	0	12	0	0	0	0	5	0	0	1	0	0	0	0	0	0
6.4	0	1	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	13	0	0
6.5	0	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
7.1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8.1	0	0	0	0	0	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	13	0	0	0
9.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9.3	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	13	0	0
9.4	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9.5	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10.1	0	10	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0
10.2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10.3	X	X	X	X	X	X	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0
10.4	X	X	X	X	X	X	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10.5	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11.1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11.2	0	16	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11.3	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Point de contrôle	Test manuel						TAW						Validator						WAI						
	Testé			Non testé			Auto			Manuel			Auto			Manuel			Auto			Manuel			
	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	A	AA	AAA	
11.4	X	X	X	X	X	X	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12.3	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
12.4	0	35	0	0	0	0	0	14	0	0	0	0	0	0	21	0	0	0	0	0	0	13	0	0	0
13.1	0	5	0	0	0	0	0	0	0	0	3	0	0	2	0	0	0	0	0	0	0	0	0	0	
13.2	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
13.3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
13.4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
13.5	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
13.6	0	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
13.7	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	
13.8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
13.9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
13.10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
14.1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14.2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
14.3	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	

Tableau 6 : Mise en relation des trois tests automatiques et du test manuel

Nous avons donc estimé l'écart entre chaque point de contrôle pour chaque application, par rapport à notre test manuel, afin de choisir celle qui possède l'écart le moins important. Il ressort de ce test que TAW est l'application se rapprochant le mieux du test manuel. Nous la choisissons donc pour tester X-TEK. Nous testons les 70 pages retenues et récupérons les rapports générés au format HTML.

### 3.3. Analyse des résultats

Après avoir testé les pages retenues, nous avons obtenu 70 rapports au format HTML. Les données de ces rapports sont cependant difficilement exploitables en l'état. Nous devons au préalable fusionner ces rapports au sein d'un document unique qui permet ensuite de réaliser une analyse des résultats ; ceci afin de faire remonter les points intéressants concernant les différents modules. Notre analyse est effectuée selon deux points de vue différents :

- Une analyse globale des erreurs, afin de faire ressortir les points communs à l'ensemble des pages.



- Une analyse par module, pour dissocier les erreurs propres à chaque module.

### 3.3.1. La fusion des rapports

Afin de rendre exploitables les données des différents rapports, nous les fusionnons dans un unique document facilement manipulable. Nous avons donc décidé de regrouper les rapports dans un document Excel afin de tirer parti des fonctions de tableau croisé dynamique d'Excel.

Il aurait été particulièrement fastidieux de retranscrire les informations des rapports à la main. C'est pourquoi nous avons rapidement développé un analyseur syntaxique en Java, qui parcourt l'ensemble des rapports présents dans un répertoire, en extrait les informations désirées et les renvoie formatées sur la sortie standard. Nous en profitons pour associer un indice au rapport testé afin de rendre la manipulation des informations plus aisée. L'indice de chaque page est donné dans le Tableau 7.

La sortie est redirigée vers un fichier texte afin d'importer les données dans Excel. Nous différencions les colonnes, à l'aide du caractère séparateur ajouté par notre application lors du formatage des données.

Pages	Indice de la page
Accueil	0
Partie administration -- copie d'un dossier	1
Partie administration -- création d'une seule page	2
Partie administration -- création d'un annuaire	3
Partie administration -- création d'un calendrier	4
Partie administration -- création d'un dossier liste avec chapitre	5
Partie administration -- création d'un dossier liste	6
Partie administration -- création d'un dossier menu	7
Partie administration -- création d'un formulaire	8
Partie administration -- création d'un forum	9
Partie administration -- création d'une liste utilisateur	10
Partie administration -- création d'une messagerie	11
Partie administration -- création d'une page web externe	12
Partie administration -- création d'une page de flux rss	13
Partie administration -- création d'un trombinoscope	14
Partie administration -- déplacement d'un dossier	15
Partie administration -- modification d'un dossier	16
TEST FRONT/Page menu déroulant/rubrique 1 - une page - hypermots	17
TEST FRONT/Page menu déroulant/rubrique 2 - une page - déroulant	18
TEST FRONT/Page menu déroulant	19
TEST FRONT/Page menu hypermots/rubrique 1 - dossier-menu - hypermots (fonction Dossier menu 1)	20

Pages	Indice de la page
TEST FRONT/Page menu hypermots/rubrique 1 - dossier-menu - hypermots (fonction Dossier menu 2 (titre masqué))	21
TEST FRONT/Page menu hypermots/rubrique 2 - dossier-menu - déroulant/Dossier menu 1	22
TEST FRONT/Page menu hypermots/rubrique 3 - Page web	23
TEST FRONT/Page menu hypermots	24
TEST FRONT/Les modules/Calendrier (Ajout d'un événement)	25
TEST FRONT/Les modules/Calendrier (Affichage d'une année)	26
TEST FRONT/Les modules/Calendrier (Affichage d'un événement)	27
TEST FRONT/Les modules/Calendrier (Affichage d'un jour)	28
TEST FRONT/Les modules/Calendrier (Recherche dans la liste d'événements)	29
TEST FRONT/Les modules/Calendrier (Affichage de la liste d'événements)	30
TEST FRONT/Les modules/Calendrier (Affichage d'une semaine)	31
TEST FRONT/Les modules/Calendrier (Affichage d'un mois)	32
TEST FRONT/Les modules/Formulaire (Consultation d'un formulaire)	33
TEST FRONT/Les modules/Formulaire (Nouveau formulaire)	34
TEST FRONT/Les modules/Formulaire	35
TEST FRONT/Les modules/Forum (Affichage du sujet de discussion 1 dans le forum 4)	36
TEST FRONT/Les modules/Forum (Affichage du forum 4)	37
TEST FRONT/Les modules/Forum (Nouveau sujet dans le forum 4)	38
TEST FRONT/Les modules/Forum (Ajout d'un forum)	39
TEST FRONT/Les modules/Forum (Recherche dans le forum 4)	40
TEST FRONT/Les modules/Forum (Réponse au sujet de discussion 1 dans le forum 4)	41
TEST FRONT/Les modules/Forum	42
TEST FRONT/Les modules/Liste utilisateur	43
TEST FRONT/Les modules/Messagerie (Affichage boîte d'envoi)	44
TEST FRONT/Les modules/Messagerie (Lire un message)	45
TEST FRONT/Les modules/Messagerie (Nouveau message)	46
TEST FRONT/Les modules/Messagerie (Affichage boîte de réception)	47
TEST FRONT/Les modules/Messagerie (Rechercher)	48
TEST FRONT/Les modules/Flux RSS	49
TEST FRONT/Les modules/Trombinoscope (Affichage en planche)	50
TEST FRONT/Les modules/Trombinoscope (Affichage d'un utilisateur)	51
TEST FRONT/Les modules/Trombinoscope	52
TEST FRONT/Page menu onglets/rubrique 1.1 - page seule standard	53
TEST FRONT/Page menu onglets/rubrique 1.2 - page seule accessibilité max	54
TEST FRONT/Page menu onglets/rubrique 1 - pages seules - hypermots	55
TEST FRONT/Page menu onglets/rubrique 2.1 - une page avec titre	56
TEST FRONT/Page menu onglets/rubrique 2.2 - une page sans titre	57
TEST FRONT/Page menu onglets/rubrique 2 - pages seules - déroulant	58
TEST FRONT/Page menu onglets/rubrique 3.2 - dossier-liste avec chap	59
TEST FRONT/Page menu onglets/rubrique 3.2 - dossier-liste avec chap (fonction Visualisation du contenu du chapitre)	60
TEST FRONT/Page menu onglets/rubrique 3.2 - dossier-liste avec chap (fonction Recherche)	61
TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste - hypermots	62
TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste – hypermots (fonction Copie d'une page)	63
TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste – hypermots (fonction Edition d'une page)	64
TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste – hypermots (fonction Déplacement d'une page)	65
TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste – hypermots (fonction Recherche)	66

Pages	Indice de la page
TEST FRONT/Page menu onglets/rubrique 3 - dossier-liste - hypermots (fonction Visualisation du contenu de "Une autre page")	67
TEST FRONT/Page menu onglets/rubrique 4.3 - dossier-liste – nav (fonction Visualisation d'une page)	68
TEST FRONT/Page menu onglets	69

**Tableau 7 : Indices des pages testées**

### 3.3.2. Les données

Nous avons obtenu, après fusion, un fichier Excel contenant 25 055 lignes d'erreurs. Chaque ligne est obtenue à partir d'un rapport TAW basé sur le WCAG 1.0. Les informations composant les lignes d'erreurs sont les suivantes :

- Le nom du rapport de la page testée
- Indice du rapport
- Le nom du module auquel appartient la page testée
- La priorité WCAG 1.0 de l'erreur
- Le point de contrôle auquel est rattachée l'erreur
- La définition de l'erreur
- L'information concernant l'identification automatique ou manuelle de l'erreur
- S'il existe, le numéro de la ligne dans le code source de la page où se situe l'erreur
- Si elle existe, la ligne de code source générant l'erreur

Bien que TAW 3.0 permette de détecter la majeure partie des erreurs, cette application est incapable de révéler toutes les erreurs comme nous avons pu le constater lors de la phase de sélection d'application de test. De plus, dans l'ensemble des erreurs retournées, l'application ne peut en identifier précisément qu'une partie. Elle divise ainsi les erreurs en deux types :

- Les erreurs, vérifiées automatiquement, que nous nommerons « erreurs automatiques ». Ce sont les erreurs dont nous pouvons être certains de l'existence.
- Les erreurs, nécessitant l'intervention d'un vérificateur humain, que nous nommerons « erreurs manuelles ».

Il est à noter qu'il existe aussi deux types d'erreurs manuelles :

- Les erreurs identifiées dans le code source de la page qui possèdent donc un numéro de ligne permettant de les repérer. Le nombre réel de ces erreurs ne peut qu'être inférieur ou égal à celui donné par l'application.
- Les erreurs à vérifier dans le document entier. Le nombre réel de ces erreurs peut être bien supérieur à celui donné par l'application.

L'application nous retourne ainsi des erreurs pour les pages testées dont certaines existent réellement et d'autres nécessitent une vérification manuelle.

Il nous faut relativiser les erreurs manuelles car nous ne pouvons pas aisément prouver leur existence puisqu'il nous faudrait les vérifier une à une pour en être certain. De plus, il ne faut pas oublier qu'il peut exister des erreurs non révélées par l'application. Bien que nous ne puissions pas les quantifier, il ne faut pas oublier qu'elles existent. Une première analyse rapide permet de donner la répartition des erreurs automatiques et manuelles pour les 70 pages testées que nous retrouvons dans le Tableau 8.

Erreurs Automatiques		Erreurs Manuelles	
Nombre d'erreurs	Pourcentage d'erreurs	Nombre d'erreurs	Pourcentage d'erreurs
8865	35.38%	16190	64.62%
		Ligne à vérifier	
		Document entier à vérifier	
Nombre d'erreurs	Pourcentage d'erreurs	Nombre d'erreurs	Pourcentage d'erreurs
14091	56.24%	2099	8.38%

**Tableau 8 : Répartition des erreurs automatiques et manuelles**

### **3.3.3. Analyse globale**

Nous passons maintenant à l'analyse globale des erreurs relevées du point de vue du WCAG 1.0. Nous commençons tout d'abord par mettre en valeur les points intéressants sur l'ensemble des pages testées. Ainsi nous pouvons voir, sur la , la répartition des erreurs selon leur niveau de priorité sur l'ensemble des pages, groupées par module. Cette figure nous permet de constater la répartition inégale des erreurs selon les modules. Les modules formulaire et calendrier possèdent ainsi le plus grand nombre d'erreurs. Nous pouvons aussi constater visuellement la prépondérance d'erreurs de priorité 2, tant manuelles qu'automatiques.

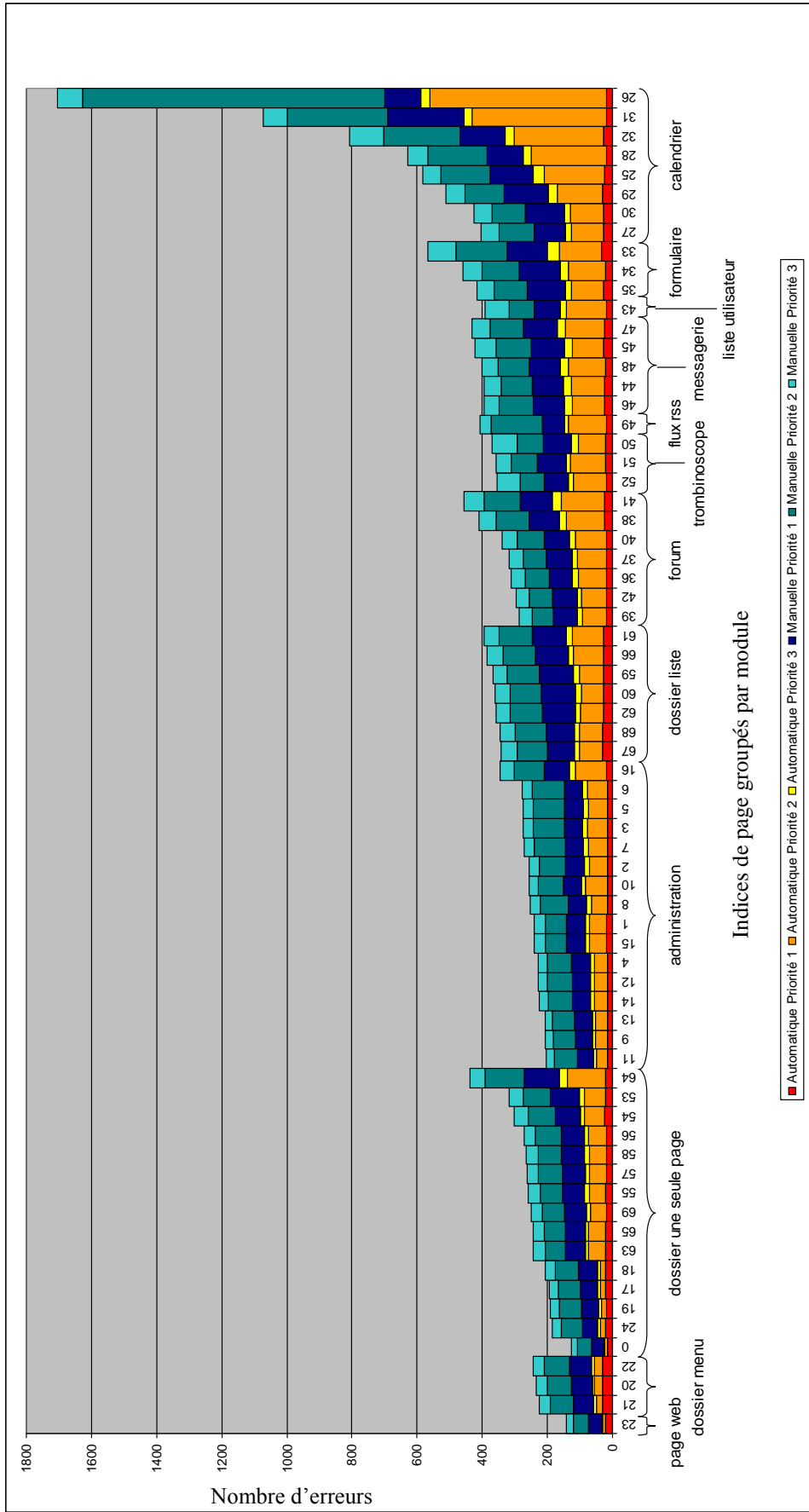


Figure 9 : Répartition des erreurs manuelles et automatiques selon leur priorité par indices de page groupés par module

Puisque nous ne possédons pas le même nombre de pages testées entre les différents modules, il est difficile de faire apparaître leur niveau de conformité. Pour ce faire, nous calculons la moyenne d'erreurs manuelles et automatiques, par priorité et par module. En prenant la moyenne d'erreurs, nous dégageons ainsi la tendance générale des modules. La Figure 10 nous montre les modules classés par moyennes d'erreurs automatiques maximales.

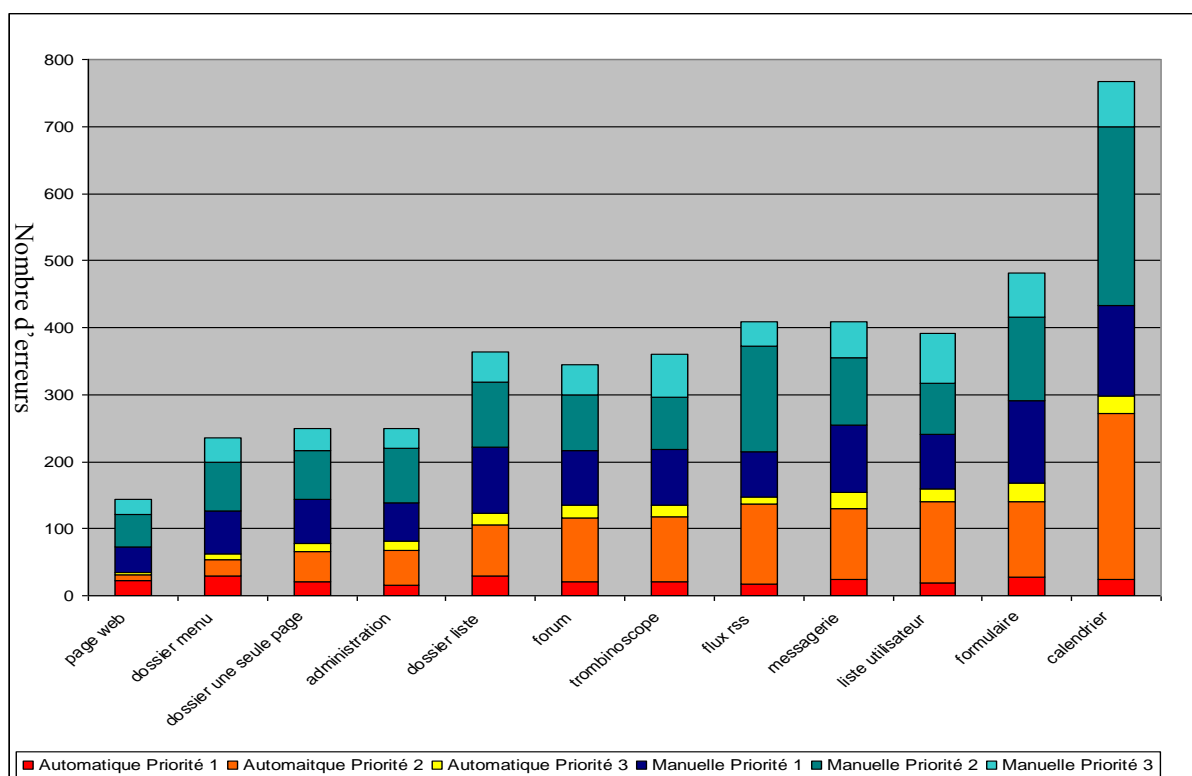


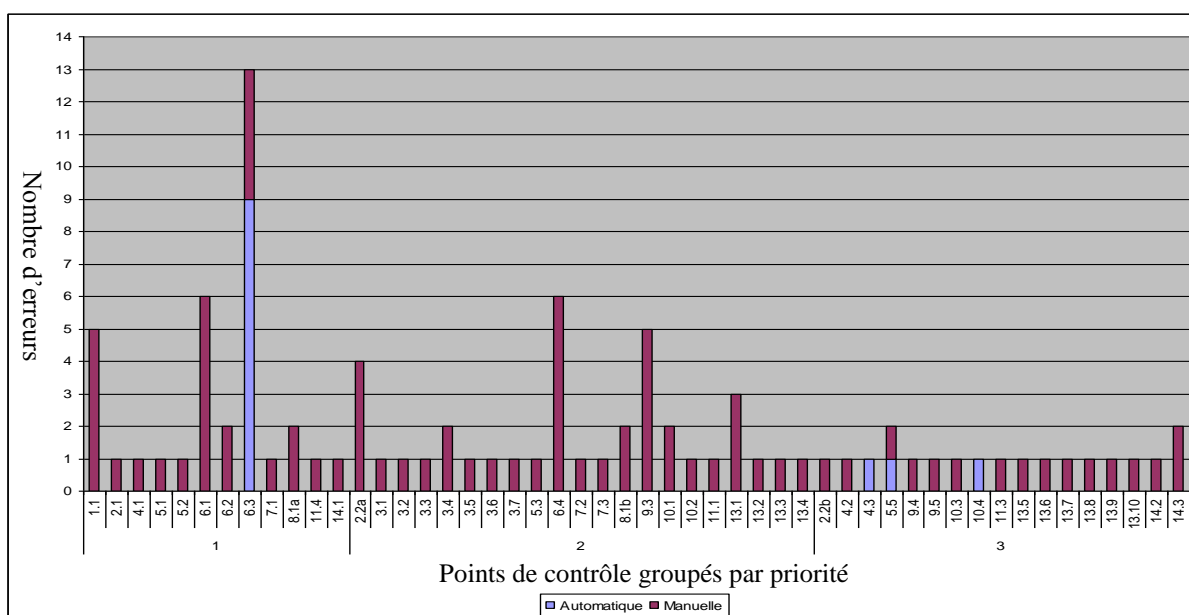
Figure 10 : Moyenne des erreurs manuelles et automatiques cumulées par priorité et par module

Nous donnons pour information, la répartition sur l'ensemble des pages, des erreurs par priorité en automatique et manuelle dans le Tableau 9. Comme nous le précisons précédemment, les erreurs de priorité 2 sont prédominantes.

Priorité 1				Priorité 2				Priorité 3			
Nombre d'erreurs		Pourcentage d'erreurs		Nombre d'erreurs		Pourcentage d'erreurs		Nombre d'erreurs		Pourcentage d'erreurs	
7256		28.96%		13567		54.15%		4232		16.89%	
Automatique		Manuelle		Automatique		Manuelle		Automatique		Manuelle	
Nombre d'erreurs	Pourcentage d'erreurs	Nombre d'erreurs	Pourcentage d'erreurs	Nombre d'erreurs	Pourcentage d'erreurs	Nombre d'erreurs	Pourcentage d'erreurs	Nombre d'erreurs	Pourcentage d'erreurs	Nombre d'erreurs	Pourcentage d'erreurs
1523	6.08%	5733	22.88%	6186	24.69%	7381	29.46%	1156	4.61%	3076	12.28%

**Tableau 9 : Répartition des erreurs automatiques et manuelles par priorité**

Enfin, nous faisons apparaître les erreurs communes à toutes les pages. La Figure 11 montre la répartition des ces erreurs pour chaque point de contrôle. Nous en profitons pour différencier les erreurs automatiques et manuelles. Nous avons ainsi 92 erreurs communes à toutes les pages qui se répartissent en 12 erreurs automatiques pour 80 manuelles. La plupart des erreurs manuelles communes nécessitent de vérifier l'ensemble de la page.



**Figure 11 : Répartition des erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

### 3.3.4. Analyse des modules

Nous analysons, dans cette partie, chaque module individuellement. Afin de faire ressortir les erreurs propres au module étudié, nous ne faisons pas apparaître les erreurs communes à

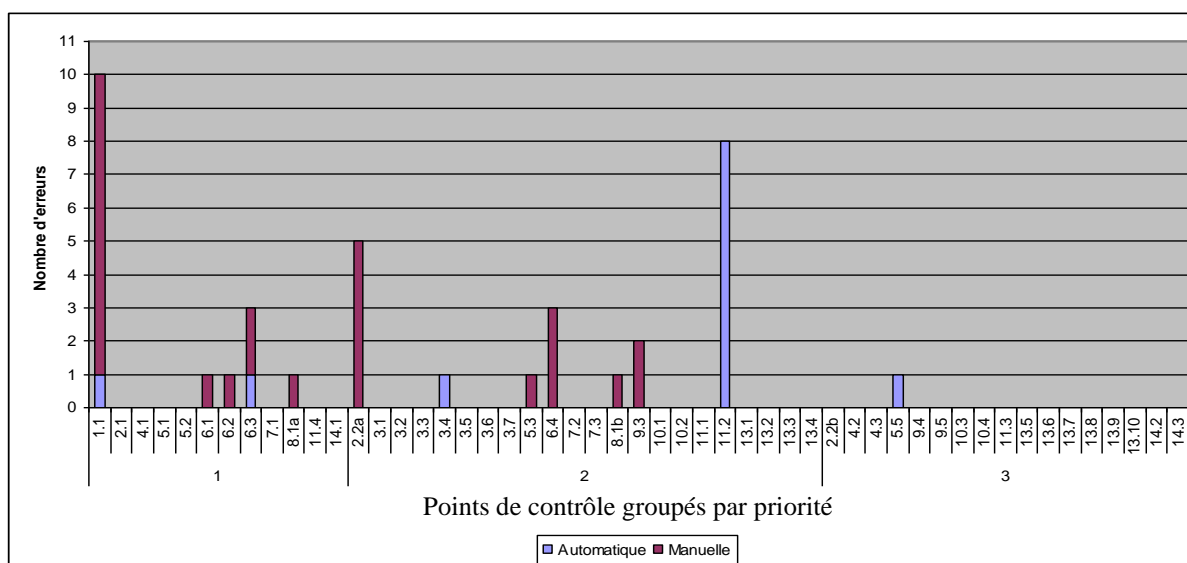


l'ensemble des pages. Nous obtenons alors des graphiques ne comportant que les erreurs du module étudié.

### 1) Analyse de la partie Administration

Nous présentons uniquement les erreurs communes à la partie Administration.

Dans la Figure 12, nous mettons en valeur la répartition des erreurs communes à la partie Administration, sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 38 erreurs communes dont 12 automatiques et 26 manuelles. La majeure partie des erreurs automatiques appartient au point de contrôle 11.2 « Ne pas utiliser de composants obsolètes ou dépréciés ». Elles sont donc liées essentiellement au respect des spécifications du W3C. Ces erreurs sont facilement corrigeables en utilisant un logiciel de validation de code HTML qui permet de signaler les différentes erreurs de respect des spécifications W3C.

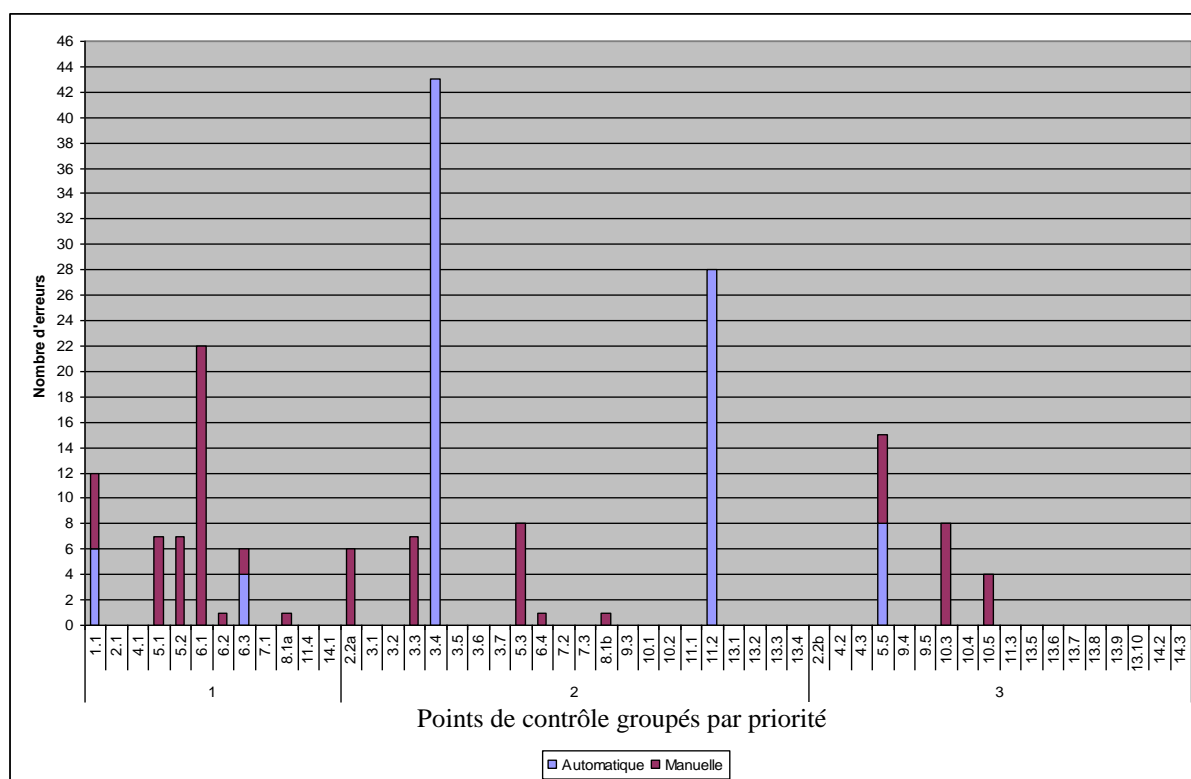


**Figure 12 : Répartition des erreurs automatiques et manuelles communes au module administration sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

### 2) Analyse du module « calendrier »

Nous présentons uniquement les erreurs communes au module « calendrier ».

Dans la Figure 13, nous mettons en valeur la répartition des erreurs communes au module « calendrier », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 177 erreurs communes dont 89 automatiques et 88 manuelles. Comme précédemment, nous avons une forte proportion des erreurs automatiques au niveau du point de contrôle 11.2. Mais nous constatons surtout une majorité d'erreurs liées au point de contrôle 3.4 « Utiliser des unités relatives pour la mise en forme ». Une partie des erreurs du point de contrôle 3.4 seront aussi facilement corrigibles en nous mettant en conformité avec la DTD<sup>24</sup> XHTML 1.0 Transitional qui est la DTD déclarée par XTEK. En effet, une partie de ces erreurs sont liées à l'usage d'unités relatives sur des balises obsolètes en XHTML. Ici encore, l'utilisation d'un logiciel de validation du code HTML facilitera la correction des erreurs.



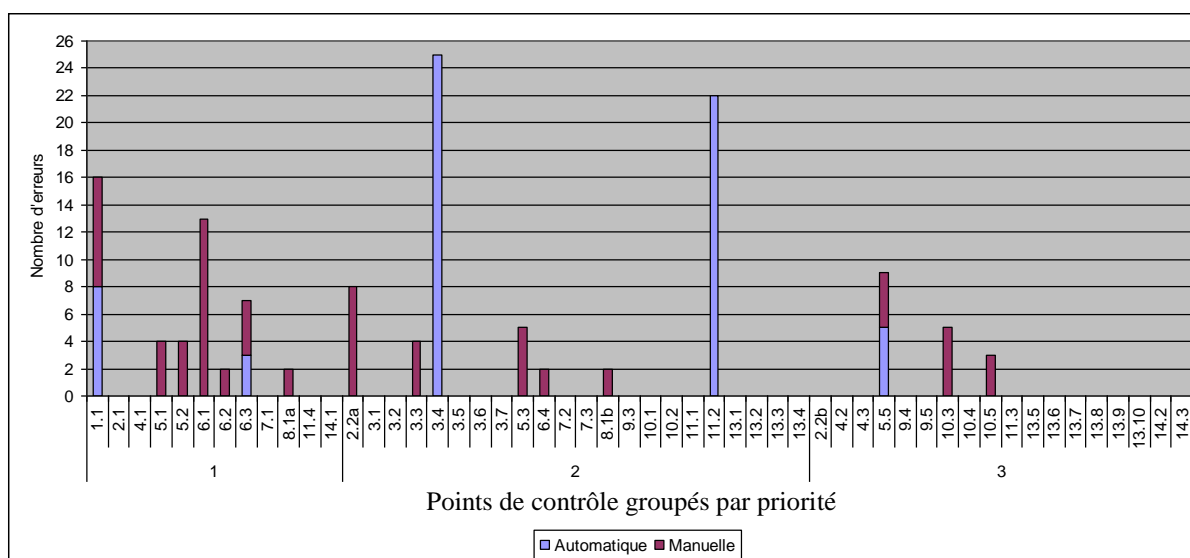
**Figure 13 : Répartition des erreurs automatiques et manuelles communes au module calendrier sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

<sup>24</sup> DTD : Document Type Definition est un document permettant de décrire le modèle de document SGML ou XML

### 3) Analyse du module « dossier liste »

Nous présentons uniquement les erreurs communes au module « dossier liste ».

Dans la Figure 14, nous mettons en valeur la répartition des erreurs communes au module « dossier liste », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 133 erreurs communes dont 63 automatiques et 70 manuelles. Nous retrouvons encore la majorité des erreurs automatiques sur les points de contrôle 3.4 et 11.2.



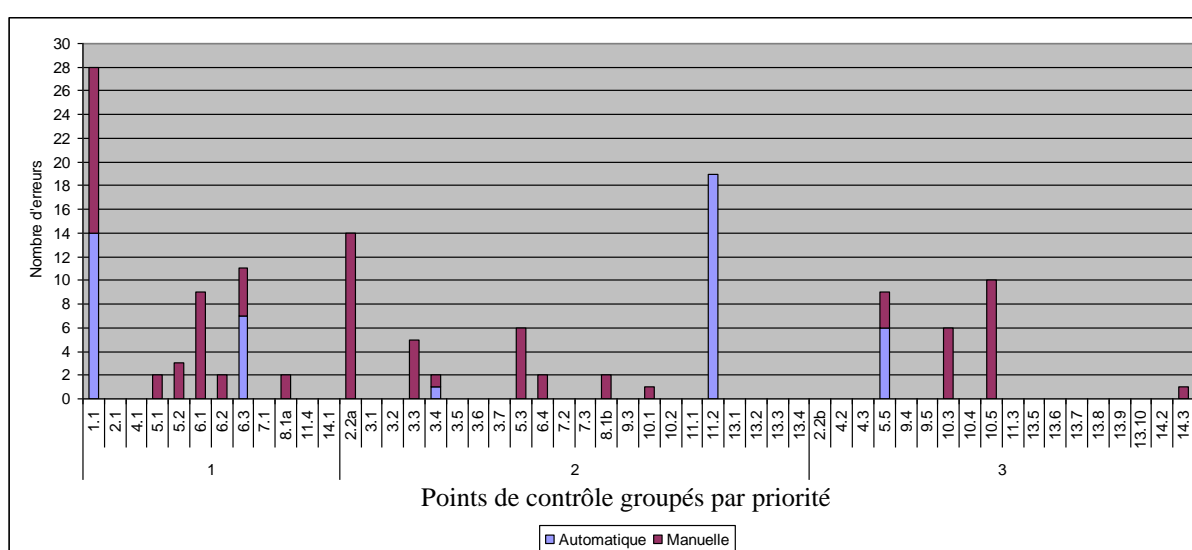
**Figure 14 : Répartition des erreurs automatiques et manuelles communes au module « dossier liste » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

### 4) Analyse du module « dossier menu »

Nous présentons uniquement les erreurs communes au module « dossier menu ».

Dans la Figure 15, nous mettons en valeur la répartition des erreurs communes au module « dossier menu », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 134 erreurs communes dont 47 automatiques et 87 manuelles. Nous relevons deux points de contrôle regroupant le maximum d'erreurs automatiques. Nous avons toujours le point de contrôle 11.2, mais nous avons aussi le 1.1 « Fournir une alternative textuelle aux

éléments non textuels ». Ce dernier concerne toutes les images et les contenus textuels à l'intérieur des images. Nous pouvons remarquer que le volume des erreurs automatiques et le volume des erreurs manuelles sur le point de contrôle 1.1 sont égaux. En effet, les erreurs automatiques reposent essentiellement sur la présence d'une alternative textuelle pour un contenu non textuel, alors que les erreurs manuelles nécessitent, quant à elles, une vérification sur la nécessité d'une description longue pour ces images. Ainsi, nous avons généralement 2 erreurs par image pour ce point de contrôle. Il est à noter aussi que les descriptions longues sont rarement utiles, ce qui implique que le nombre réel d'erreurs manuelles pour ce point de contrôle est souvent nul.



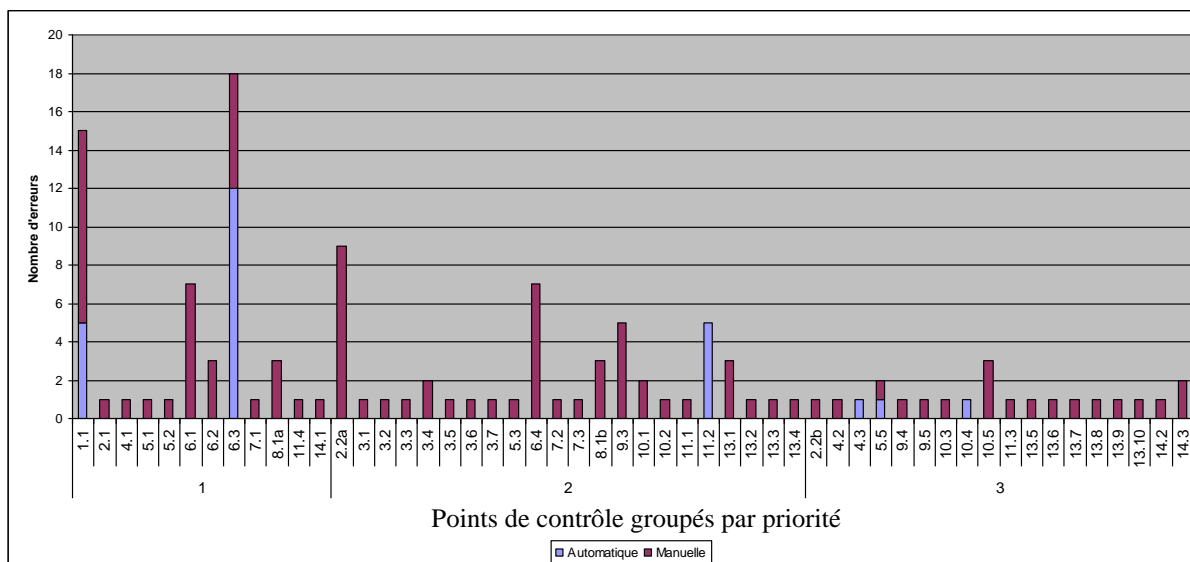
**Figure 15 : Répartition des erreurs automatiques et manuelles communes au module « dossier menu » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

### 5) Analyse du module « dossier une seule page »

Nous présentons uniquement les erreurs communes au module « dossier une seule page ». Il est à noter que nous ne prenons pas en compte la page d'accueil qui semble avoir un comportement à part des autres pages « dossier une seule page ». En effet, elle s'approche beaucoup trop des erreurs communes à l'ensemble des pages, ce qui n'a pas d'intérêt pour nous et occulterait les points communs à toutes les autres pages.

Dans la Figure 16, nous mettons en valeur la répartition des erreurs communes au module « dossier une seule page », sans les erreurs communes à l'ensemble des pages, en les groupant

par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 125 erreurs communes dont 25 automatiques et 100 manuelles. Les erreurs portent essentiellement sur les points 1.1 et 6.3 « Proposer une alternative au code javascript et aux éléments programmables »



**Figure 16 : Répartition des erreurs automatiques et manuelles communes au module « dossier une seule page » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité (page d'accueil exclu)**

Nous avons créé, pour cette analyse, deux pages particulières utilisant toutes les possibilités de l'éditeur de texte enrichi de l'outil X-TEK. La page d'indice « 53 » a été conçue sans aucun souci quant à son accessibilité alors que celle d'indice « 54 » a été conçue en exploitant toutes les capacités d'accessibilité de l'éditeur.

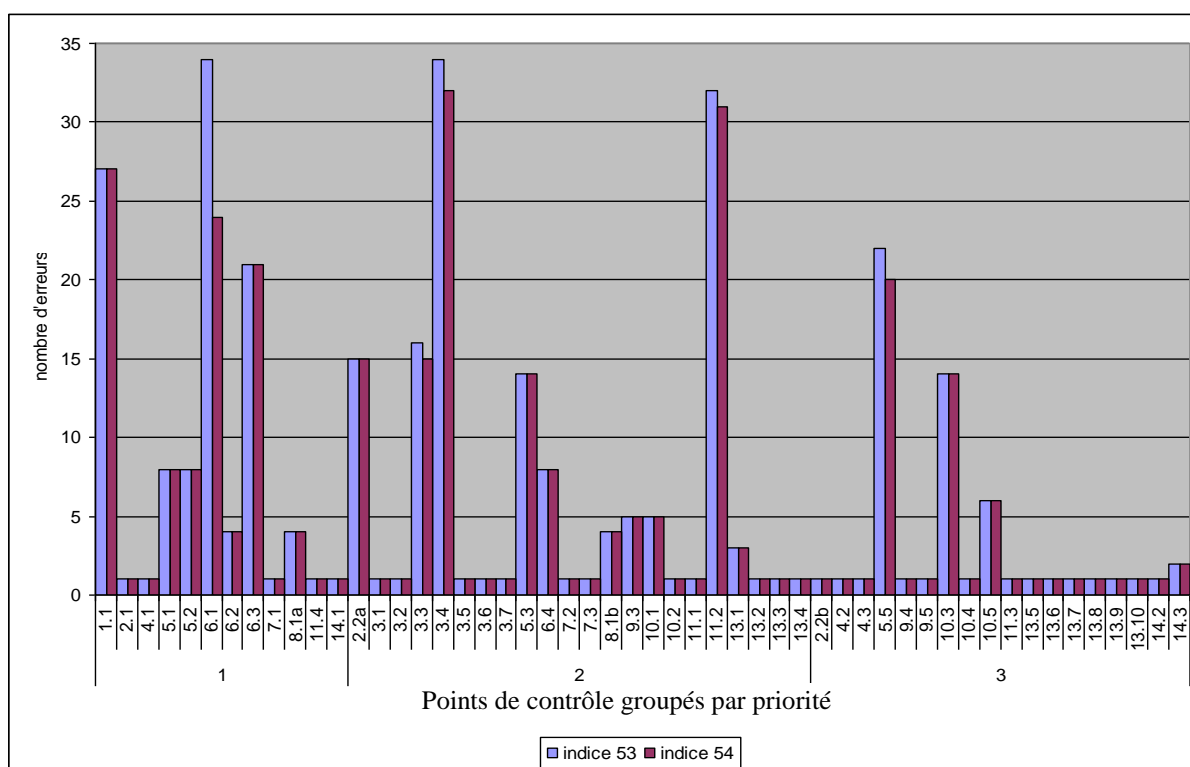
Nous obtenons ceci :

- Pour la page d'indice 53, nous avons relevé 317 erreurs qui se répartissent en 103 automatiques et 214 manuelles.
- Pour la page d'indice 54, nous avons relevé 301 erreurs qui se répartissent en 100 automatiques et 201 manuelles.

Au final, il n'existe qu'un gain de 16 erreurs avec une page utilisant toutes les possibilités d'accessibilité de l'éditeur.

La Figure 17, nous montre la répartition des erreurs pour ces deux pages par point de contrôle. Nous constatons uniquement des gains sur les points de contrôle suivants :

- 6.1 : Maintenir la lisibilité et la compréhension des contenus lorsque les styles sont désactivés
- 3.3 : Privilégier l'utilisation des styles CSS pour la mise en forme
- 3.4 : Utiliser des unités relatives pour la mise en forme
- 11.2 : Ne pas utiliser de composants obsolètes ou dépréciés
- 5.5 : Donner des informations complémentaires sur les tableaux de données



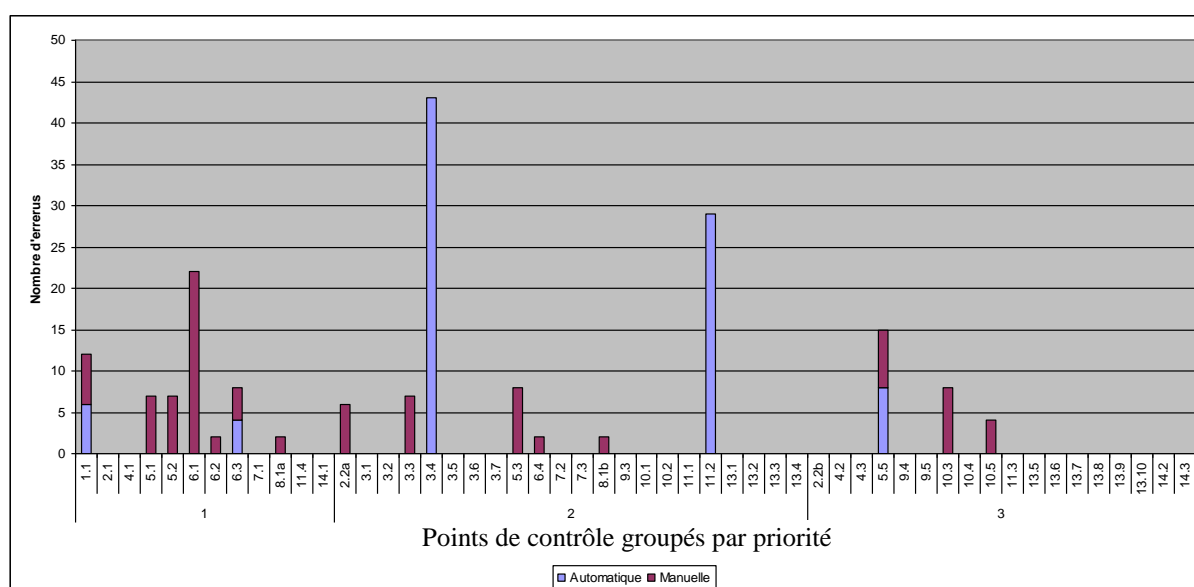
**Figure 17 : Répartition des erreurs pour les pages d'indice 53 et 54, par points de contrôle groupés par priorité**

En sachant que ces pages possèdent un contenu identique, le gain obtenu n'est pas très important. L'éditeur de texte enrichi posera d'emblée de nombreux problèmes quant au respect des normes d'accessibilité.

## 6) Analyse du module « formulaire »

Nous présentons uniquement les erreurs communes au module « formulaire ».

Dans la Figure 18, nous mettons en valeur la répartition des erreurs communes au module « formulaire », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 184 erreurs communes dont 90 automatiques et 94 manuelles. Comme pour les modules « dossier liste » et calendrier, les points de contrôle comportant le plus d'erreurs sont les points 3.4 et 11.2. Nous en tirons les mêmes conclusions, c'est-à-dire la possibilité de corriger ces erreurs à l'aide d'un logiciel de validation du code HTML.



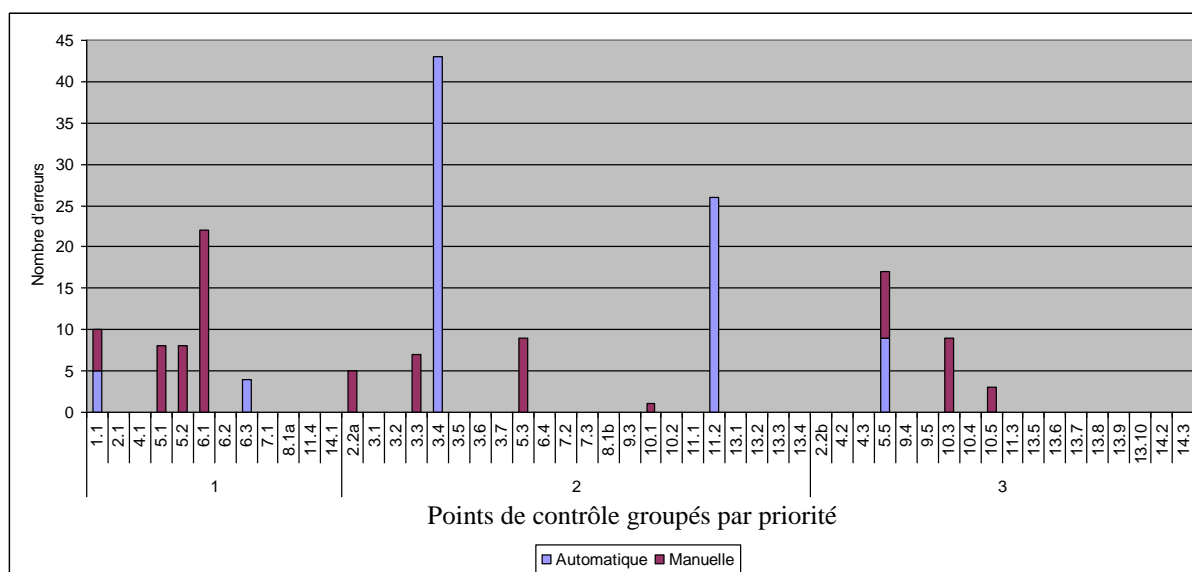
**Figure 18 : Répartition des erreurs automatiques et manuelles communes au module formulaire sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

## 7) Analyse du module « forum »

Nous présentons uniquement les erreurs communes au module « forum ».

Dans la Figure 19, nous mettons en valeur la répartition des erreurs communes au module « formulaire », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons

ainsi 172 erreurs communes dont 87 automatiques et 85 manuelles. Nous avons toujours une majorité d'erreurs sur les points de contrôle 3.4 et 11.2.



**Figure 19 : Répartition des erreurs automatiques et manuelles communes au module forum sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

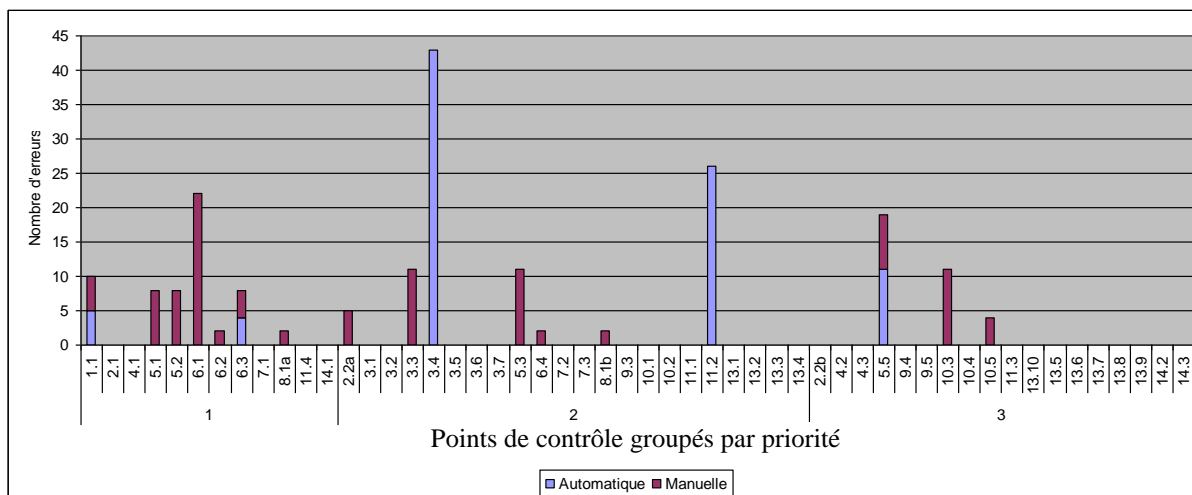
## 8) Analyse du module « messagerie »

Nous présentons uniquement les erreurs communes au module « messagerie ».

Dans la Figure 20, nous mettons en valeur la répartition des erreurs communes au module « messagerie », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 194 erreurs communes dont 89 automatiques et 105 manuelles. Comme sur le module précédent, nous avons aussi une majorité d'erreurs sur les points de contrôles 3.4 et 11.2. Nous avons aussi des erreurs sur le point de contrôle 5.5 « Donner des informations complémentaires sur les tableaux de données » liées à l'utilisation de tableaux à des fins de présentation. TAW ne pouvant pas faire la différence entre un tableau de données et un tableau de présentation, il signale alors des erreurs comme l'absence de titre, de résumé de tableau et bien d'autres. La correction de la structure de la page, en passant d'une présentation par tableau à une présentation entièrement conçue à l'aide d'une feuille de style, permettra de résoudre ces erreurs. Nous avons aussi des erreurs sur le point de contrôle 6.1 « Maintenir la



lisibilité et la compréhension des contenus lorsque les styles sont désactivés » liées à l'usage des styles dans le code source. TAW nous demande de vérifier la lisibilité de la page lorsque les styles sont désactivés. Il nous faudra passer l'ensemble des styles du code source vers les feuilles de style.

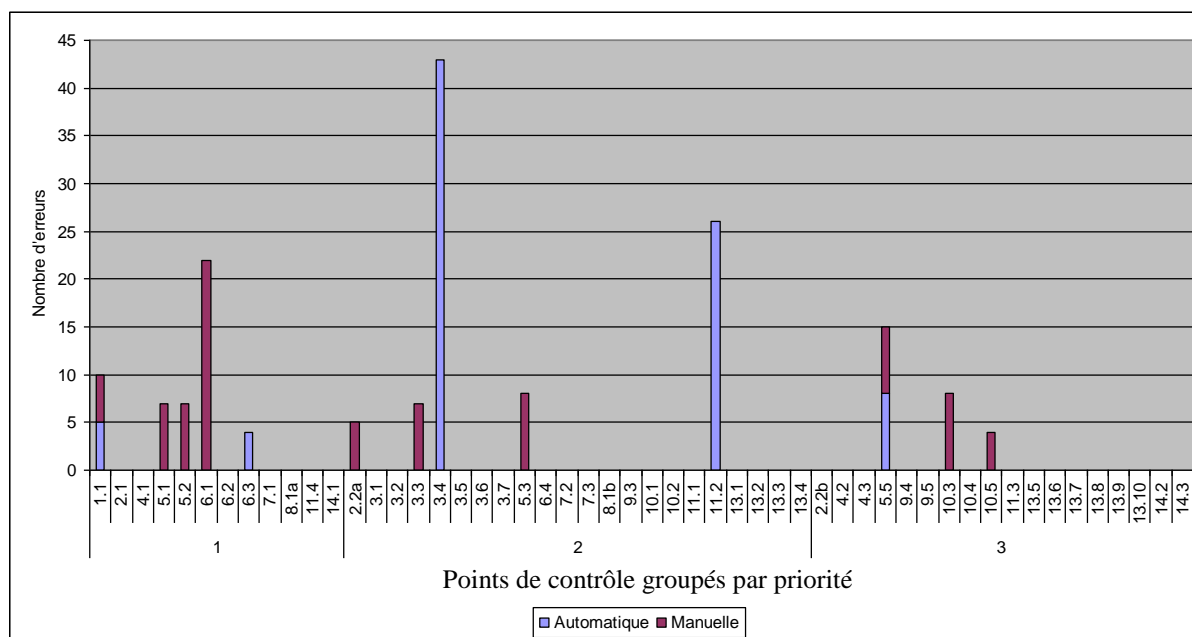


**Figure 20 : Répartition des erreurs automatiques et manuelles communes au module messagerie sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

## 9) Analyse du module « trombinoscope »

Nous présentons uniquement les erreurs communes au module « trombinoscope ».

Dans la Figure 21, nous mettons en valeur la répartition des erreurs communes au module « trombinoscope », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 166 erreurs communes dont 86 automatiques et 80 manuelles. Comme pour le module « messagerie », nous avons essentiellement des erreurs sur les points de contrôle 3.4, 11.2, 5.5 et 6.1.



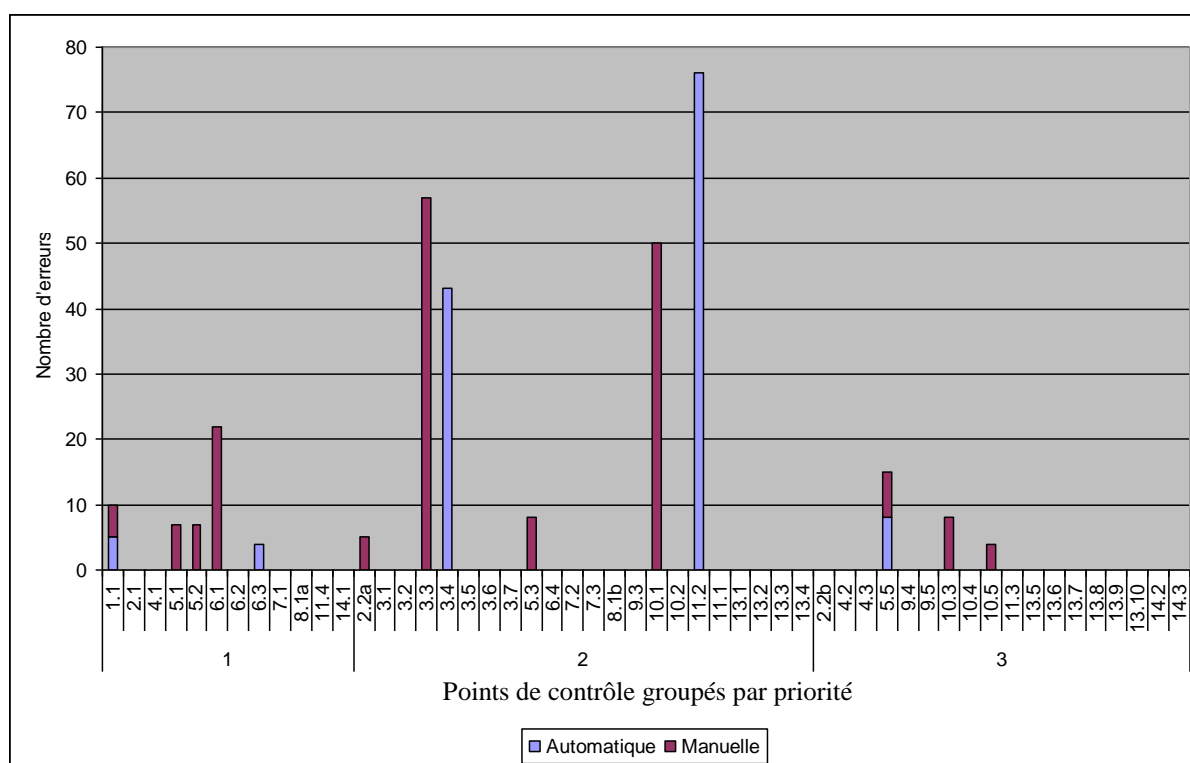
**Figure 21 : Répartition des erreurs automatiques et manuelles communes au module trombinoscope sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

## 10) Analyse du module « flux rss »

Nous présentons uniquement les erreurs communes au module « flux rss ». Il faut cependant prendre en compte le fait que nous ne possédons qu'une seule page testée pour ce module, ce qui ne permet aucunement de déduire quoi que ce soit concernant les erreurs communes à ce module.

Dans la Figure 22, nous mettons en valeur la répartition des erreurs communes au module « flux rss », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 316 erreurs communes dont 136 automatiques et 180 manuelles. La majorité des erreurs se situe sur les points de contrôle 11.2, 3.4, 3.3 « Privilégier l'utilisation des styles CSS pour la mise en forme » et 10.1 « Signaler l'ouverture de nouvelles fenêtres ». Les erreurs 3.3 sont dues à l'utilisation de balises HTML de mise en forme dépréciées telles que la balise « font » et des attributs tels que « color ». Ces erreurs sont aisément corrigibles en passant entièrement la mise en forme en feuilles de style et en faisant disparaître les balises et attributs dépréciés. Les erreurs du point de contrôle 10.1 sont dues aux liens des flux RSS qui, pour permettre la consultation des pages qu'ils pointent, ouvrent une nouvelle fenêtre. En faisant en

sorte de fournir une information sur lien avertissant de l'ouverture d'une nouvelle fenêtre, nous apporterons une solution à ces erreurs.



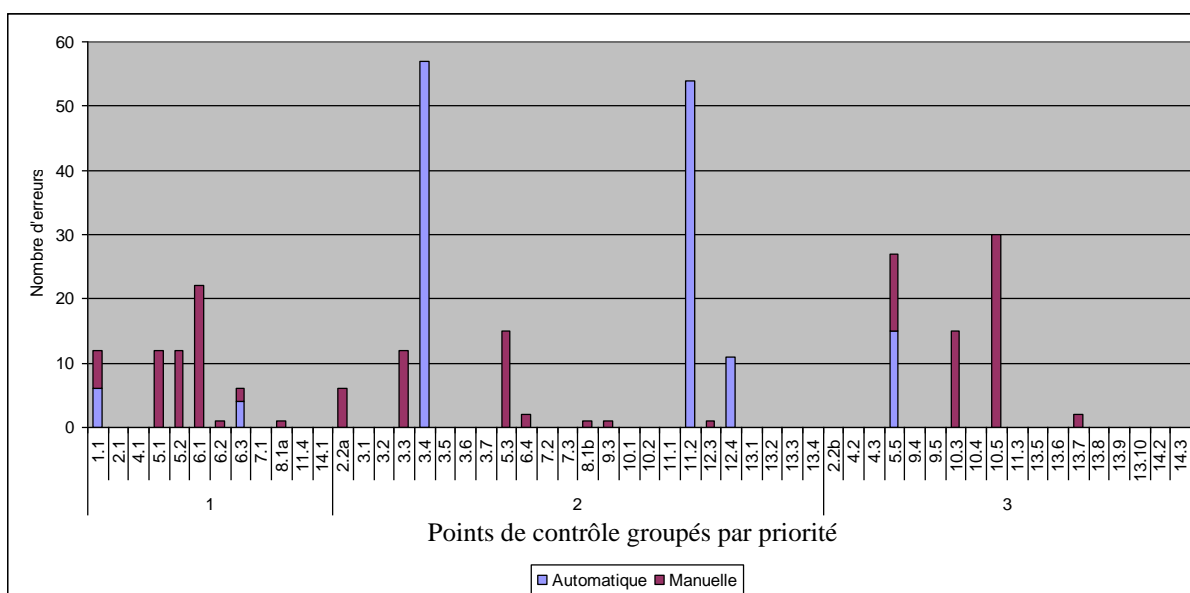
**Figure 22 : Répartition des erreurs automatiques et manuelles communes au module « flux rss » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

### 11) Analyse du module « liste des utilisateurs »

Nous présentons uniquement les erreurs communes au module « liste des utilisateurs ». Il faut cependant prendre en compte que nous ne possédons qu'une seule page testée pour ce module, ce qui ne permet aucunement de déduire quoi que ce soit concernant les erreurs communes à ce module.

Dans la Figure 23, nous mettons en valeur la répartition des erreurs communes au module « liste utilisateur », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 300 erreurs communes dont 147 automatiques et 153 manuelles. Nous retrouvons encore la répartition des erreurs sur les points de contrôle 3.4, 11.2 et 5.5. Nous rencontrons aussi des erreurs sur le point de contrôle 10.5 « Séparer visuellement les liens adjacents ». La

solution préconisée pour résoudre ces erreurs est d'introduire entre les liens, un caractère imprimable n'appartenant pas à un lien, entouré d'espaces pour séparer visuellement les liens.

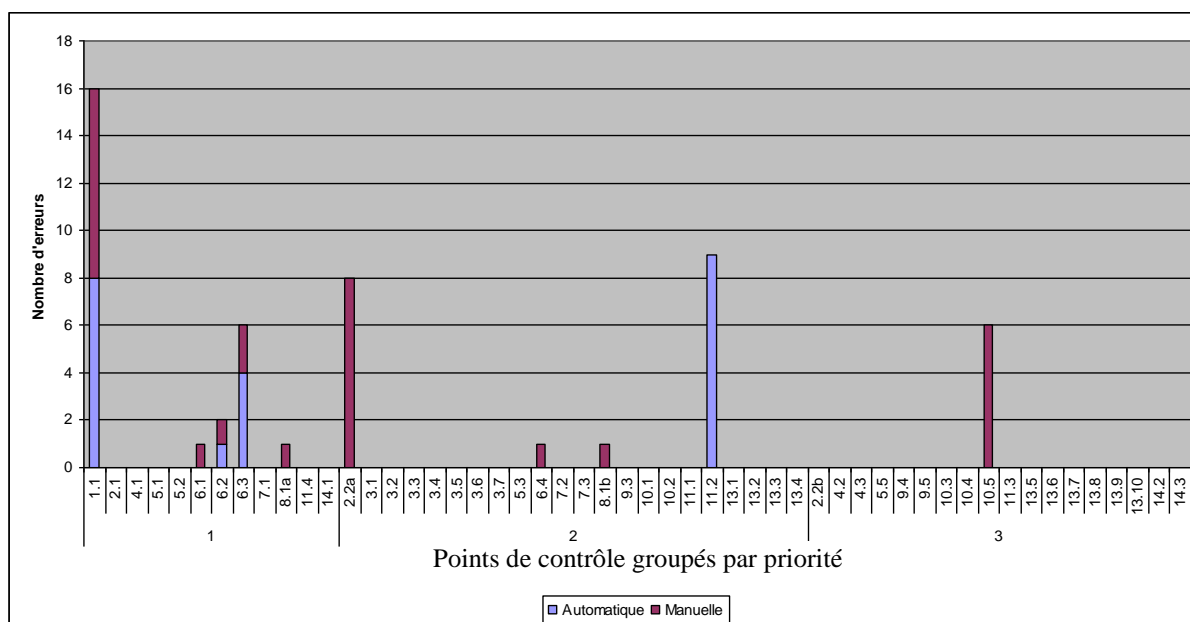


**Figure 23 : Répartition des erreurs automatiques et manuelles communes au module « liste des utilisateurs » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

## 12) Analyse du module « page web »

Nous présentons uniquement les erreurs communes au module « page web ». Il faut cependant prendre en compte que nous ne possédons qu'une seule page testée pour ce module, ce qui ne permet aucunement de déduire quoi que ce soit concernant les erreurs communes à ce module.

Dans la Figure 24, nous mettons en valeur la répartition des erreurs communes au module « page web », sans les erreurs communes à l'ensemble des pages, en les groupant par point de contrôle et en différenciant les erreurs automatiques des erreurs manuelles. Nous avons ainsi 51 erreurs communes dont 22 automatiques et 29 manuelles. Nous avons une majorité d'erreurs pour les points de contrôle 1.1 et 11.2.



**Figure 24: Répartition des erreurs automatiques et manuelles communes au module « page web » sans les erreurs automatiques et manuelles communes à toutes les pages, par points de contrôle groupés par priorité**

### 3.4. Synthèse

Après cette analyse des données, nous allons maintenant synthétiser ces informations afin d'en extraire les points les plus importants, pour préconiser des solutions d'amélioration de l'accessibilité. Nous commencerons par exploiter les résultats de l'analyse et nous donnerons nos conclusions.

#### 3.4.1. Exploitation

Nous avons mis en valeur, précédemment, des informations intéressantes concernant les erreurs communes à chaque module, sans les erreurs communes à l'ensemble des pages et les erreurs communes à l'ensemble des pages testées. Dans le Tableau 10, nous mettons en valeur la part des erreurs communes dans l'ensemble des erreurs des pages testées. On peut constater aisément que les 92 erreurs communes à toutes les pages représentent à elles seules 25% de l'ensemble des erreurs des pages testées. De plus, l'ensemble des erreurs communes, soit 2099, représentent 61% des erreurs des pages testées.

		Nombre d'erreurs total		25055	
Erreurs communes		Nombre de pages	Nombre d'erreurs communes	Nombre d'erreurs communes total	Pourcentage d'erreurs communes sur le nombre d'erreurs total
page web		1	51	51	0.20%
dossier menu		3	134	402	1.60%
dossier une seule page (accueil exclu)		14	125	1750	6.98%
accueil		1	17	17	0.07%
administration		16	38	608	2.43%
dossier liste		7	133	931	3.72%
forum		7	172	1204	4.81%
trombinoscope		3	166	498	1.99%
flux rss		1	316	316	1.26%
messagerie		5	194	970	3.87%
liste des utilisateurs		1	300	300	1.20%
formulaire		3	184	552	2.20%
calendrier		8	177	1416	5.65%
ensemble des pages		70	92	6440	25.70%
<b>Total</b>			2099	15455	61.68%

**Tableau 10 : Part des erreurs communes dans l'ensemble des erreurs**

Dans le Tableau 11, nous nous intéressons à la part des erreurs communes automatiques et manuelles dans l'ensemble des erreurs automatiques et des erreurs manuelles. Nous pouvons constater que les erreurs communes automatiques génèrent 50% des erreurs automatiques trouvées et les erreurs communes manuelles génèrent, quant à elles, 67% des erreurs manuelles trouvées. Il est bon de constater que les erreurs communes manuelles de tous les modules représentent, à elles seules, la moitié des erreurs communes manuelles dans les pages testées. De plus, elles regroupent la plupart des erreurs manuelles nécessitant une analyse totale des pages. On peut en déduire que les erreurs restantes seront essentiellement composées d'erreurs déjà positionnées, donc plus faciles à traiter.

	Erreurs automatiques				Erreurs manuelles		
	Nombre d'erreurs total	8865			Nombre d'erreurs total	16190	
Le module	Nombre de pages	Nombre d'erreurs communes	Nombre d'erreurs communes total	Pourcentage d'erreurs communes sur le nombre d'erreurs total	Nombre d'erreurs communes	Nombre d'erreurs communes total	Pourcentage d'erreurs communes sur le nombre d'erreurs total
page web	1	22	22	0.25%	29	29	0.18%
dossier menu	3	47	141	1.59%	87	261	1.61%
dossier une seule page (accueil exclu)	14	25	350	3.95%	100	1400	8.65%
accueil	1	15	15	0.17%	18	18	0.11%
administration	16	12	192	2.17%	26	416	2.57%
dossier liste	7	63	441	4.97%	70	490	3.03%
forum	7	87	609	6.87%	85	595	3.68%
trombinoscope	3	86	258	2.91%	80	240	1.48%
flux rss	1	136	136	1.53%	180	180	1.11%
messagerie	5	89	445	5.02%	105	525	3.24%
liste des utilisateurs	1	147	147	1.66%	153	153	0.95%
formulaire	3	90	270	3.05%	94	282	1.74%
calendrier	8	89	712	8.03%	88	704	4.35%
ensemble des pages	70	12	840	9.48%	80	5600	34.59%
<b>Total</b>		920	4578	51.64%	1195	10893	67.28%

**Tableau 11:Part des erreurs communes automatiques et manuelles dans l'ensemble des erreurs automatiques et manuelles**

Nous pouvons maintenant mettre en valeur la part des erreurs communes dans l'ensemble des erreurs par module. Pour ce faire, nous additionnons aux erreurs communes propres au module, les erreurs communes à l'ensemble des pages comme nous pouvons le voir dans le Tableau 12. Nous ne tenons pas compte des modules « page web », « flux rss » et « liste des utilisateurs » qui, ne comportant qu'une seule page dans notre test, donnent un rapport de 100%. Nous avons de même exclu la page d'accueil pour son fonctionnement particulier. Nous pouvons cependant constater que les erreurs communes représentent en général plus de 50% des erreurs des modules à l'exception du module calendrier. Il est intéressant de noter que le calendrier représente une grande part des erreurs mais ses erreurs communes ne correspondent qu'à 35% des erreurs du module. Ce module demandera donc le plus de travail pour le rendre accessible.

Le module	nombre de pages	Nombre d'erreur commune du module avec les erreurs communes à l'ensemble des pages	Nombre total d'erreurs communes du module	Nombre d'erreurs total des modules	Rapport des erreurs communes sur l'ensemble des erreurs des modules
dossier menu	3	226	678	706	96%
dossier une seule page (accueil exclu)	14	217	3038	3753	81%
administration	16	130	2080	3984	52%
dossier liste	7	225	1575	2550	62%
forum	7	264	1848	2417	76%
trombinoscope	3	258	774	1082	72%
messagerie	5	286	1430	2044	70%
formulaire	3	276	828	1442	57%
calendrier	8	269	2152	6134	35%

**Tableau 12:part des erreurs communes dans l'ensemble des erreurs des modules**

### 3.4.2. Conclusions

Bien que nous ayons relevé 25055 erreurs pour 70 pages testées, soit 358 erreurs par page, la tâche de mise en conformité n'est pas insurmontable. En effet, l'outil X-TEK étant de conception modulaire, nous avons pu découper l'analyse en nous axant sur les modules. Nous avons pu mettre en évidence les erreurs communes aux pages des modules et les erreurs communes à l'ensemble des modules qui correspondent, en grande partie, à la charte graphique. Nous en avons déduit qu'en corrigeant les 2099 erreurs communes identifiées, nous pouvons corriger 61% des 25055 erreurs relevées.

Nous avons aussi identifié le pourcentage d'erreurs que nous pouvons corriger dans chaque module, dans un premier temps, en nous focalisant uniquement sur les erreurs communes. Par la même, nous avons pu identifier les modules demandant le plus de travail pour être mis en conformité.

Enfin, nous avons aussi pu constater le problème de génération de contenu accessible de l'éditeur de texte enrichi embarqué dans X-TEK. Il est indéniable qu'il faudra trouver un nouvel éditeur afin de garantir l'accessibilité des plateformes générées.



### 3.5. Recommandations générales

Après cette longue étude, nous avons formulé quatre recommandations :

- Le nombre d'erreurs identifiées étant assez important, il nous est impossible, dans un temps raisonnable, de traiter toutes ces erreurs. Cependant, nous avons pu identifier 2099 erreurs communes qui, une fois traitées, permettraient de lever plus de 60% des erreurs identifiées. Nous recommandons d'effectuer une première phase afin de préparer le travail de mise en conformité et de rendre l'identification des erreurs restantes plus facile. Pour ce faire, nous devons traiter les 2099 erreurs communes et corriger toutes les erreurs que nous identifierons au cours de cette phase. Nous en profiterons pour valider les pages en XHTML Transitional.
- Nous avons pu remarquer que l'éditeur de texte enrichi ne permet pas de générer du contenu accessible, ce qui est particulièrement gênant, vu que la majeure partie des pages contenues sur les sites générés par X-TEK sont produites à l'aide de l'éditeur de texte enrichi. Nous recommandons la recherche d'un nouvel éditeur de texte enrichi permettant la production de pages accessibles.
- Même si nous fournissons un éditeur de texte enrichi conforme à nos attentes, l'accessibilité repose en grande partie sur le contenu et le texte alternatif aux contenus non textuels. Il faudra que chaque contributeur s'investisse dans l'accessibilité et produise du contenu cohérent et accessible. Malgré la bonne volonté de chacun, nous ne serons pas à l'abri de problèmes d'accessibilité du contenu. Il faut donc mettre en place une solution permettant de valider le contenu des contributeurs. Nous recommandons d'utiliser le système de Workflow<sup>25</sup> existant dans X-TEK afin que les pages qui devront être accessibles soient validées par un responsable « accessibilité ».
- La structure des pages utilise déjà les feuilles de style pour leur mise en forme. Cependant, leur usage n'est pas optimal. X-TEK utilise encore de nombreux tableaux de présentation, des définitions de style en ligne, des balises de mise en forme

---

<sup>25</sup> Workflow : Un Workflow, ou en français « flux de travaux », est une technologie visant à informatiser les diverses étapes d'un processus ou la circulation des documents.

dépréciées, etc. Nous recommandons de revoir entièrement l'usage des feuilles de style afin de séparer totalement la mise en forme du contenu.

### **3.6. Définition du plan de mise en conformité**

Afin de définir le plan de mise en conformité de l'application, nous avons organisé une réunion avec les principaux acteurs et décideurs du projet X-TEK que sont :

- Monsieur RAMEAU Gérard : chef du projet et s'occupant de la conception, de l'ergonomie, de la maîtrise d'œuvre, des relations institutionnelles et du déploiement
- Monsieur VASSOILLE Marc : Responsable développement informatique et conception
- Monsieur GIRAUDIN Eric : Intervenant au niveau de la conception et de l'aspect graphique

Nous avons aussi pu compter parmi nous Madame PESTY Sylvie, enseignante chercheuse à l'IUT2, qui sensibilise les étudiants de l'IUT à l'accessibilité lors de son cours sur les technologies Web.

Après un rappel sur les normes d'accessibilité et la présentation de notre étude, nous avons exposé nos recommandations qui ont fait l'unanimité parmi les membres de cette réunion.

Toutefois, les membres du projet X-TEK se sont tout de suite entendus pour que nous nous concentrons principalement, dans une première phase, sur la charte graphique et le module « dossier une seule page » qui sont prioritaires pour les plateformes Webtek. En effet, la majorité des pages composant un Webtek sont constituées de « dossier une seule page ». Les grandes parties des autres modules ne sont en l'occurrence disponibles qu'à partir du BackOffice de l'application. Nous devons faire en sorte que le Front-Office soit le plus rapidement accessible. Nous apprendrons par la suite que plusieurs modules que nous croyions distincts sont en fait intimement liés au module dossier une seule page. Ceci n'aurait pu être mis en valeur sans rentrer dans le code source de l'application.

Le plan de mise en conformité dans cette première phase est donc le suivant :

1. Mettre en conformité la charte graphique en se focalisant dans un premier temps sur la page d'accueil.
2. Mettre en conformité l'ensemble du module « dossier une seule page ».
3. Présenter les résultats obtenus au cours d'une réunion

### **3.7. Conclusion**

Nous avons réalisé dans ce chapitre un audit d'accessibilité de X-TEK. Nous nous sommes basés sur le RGAA 1.0, et donc sur le WCAG 1.0, qui était le référentiel officiel lors de la réalisation de cet audit.

Au cours de cette analyse portant sur 70 pages et ayant permis d'identifier 25055 erreurs, nous avons pu mettre en valeur la présence de 92 erreurs communes à l'ensemble des pages testées représentant 25% de l'ensemble des erreurs identifiées. Nous avons aussi pu faire ressortir que les erreurs communes à l'ensemble des pages testées et les erreurs communes à chaque module, soit 2099 erreurs, comptent pour 61% de l'ensemble des erreurs identifiées.

De plus, nous avons montré que l'éditeur de texte enrichi actuel ne permet pas la production de contenu accessible. Nous avons aussi souligné le fait que même si nous parvenions à mettre en place un éditeur de texte enrichi produisant du contenu accessible, une grande part de l'accessibilité passerait par la formation des contributeurs.

A partir de cette analyse, nous avons établi les quatre recommandations suivantes :

- Effectuer une première phase de mise en conformité portant principalement sur les 2099 erreurs communes identifiées.
- Rechercher un éditeur de texte enrichi produisant du contenu accessible.
- Mettre en place un mécanisme permettant de faire valider, par un responsable accessibilité, les pages produites par les contributeurs.
- Séparer totalement la mise en forme du contenu à l'aide de feuilles de style.

Enfin, nous avons présenté nos recommandations lors de la réunion de définition du plan de mise en conformité. Bien qu'ayant fait l'unanimité, l'équipe X-TEK a décidé de focaliser nos efforts sur le module « dossier une seule page ».

Dans le chapitre suivant, nous aborderons la mise en conformité du module « dossier une seule page ».



## 4. Mise en conformité

Nous allons, dans un premier temps, présenter le fonctionnement de X-TEK afin de donner une vue plus concrète des parties de l'application sur lesquelles nous allons travailler. Puis nous présenterons le déroulement de la mise en conformité suivi de l'application des tests RGAA 2.2. Ensuite, nous aborderons notre recherche d'un nouveau éditeur de texte enrichi et nous finirons par la présentation de nos résultats, lors de la réunion marquant la fin de la mise en conformité du module « dossier une seule page » de X-TEK.

### 4.1. Étude du fonctionnement de X-TEK

Afin de fournir une vue plus concrète de l'application, nous présentons brièvement le fonctionnement de X-TEK à travers la partie graphique et le code source. Nous terminons par une présentation du patron de conception rencontré lors de la mise en conformité.

#### 4.1.1. Partie graphique

La partie graphique de X-TEK repose sur l'utilisation de templates et de feuilles de styles. Pour mettre en œuvre son système de templates, X-TEK utilise le moteur de Template Smarty en version 2.6.7. La structure de la page s'articule autour des 4 templates « haut.tpl », « bas.tpl », « menu.tpl » et « menua.tpl ».

Le template « haut.tpl » est appelé en premier et gère la barre d'outils, les rubriques, les sous-rubriques et la barre de boutons d'administration. Le template « bas.tpl » est appelé après la zone dynamique et gère le logo, l'appel aux différents menus, les publications, la recherche et la zone de pied de page. Le template « menu.tpl » gère le menu des pages standard tandis que le template « menua.tpl » gère le menu de la partie administration de la plateforme. La Figure 25 décrit le découpage type des différents templates. Les données des templates sont générées par l'application et transférées aux templates en initialisant des variables utilisées par Smarty. Le contenu propre de la page n'est pas géré par les templates et est directement introduit entre le template « haut.tpl » et « bas.tpl » dans la zone dynamique. Le code HTML est, dans ce cas, produit à partir du code PHP et/ou récupéré en base de données. La mise en forme de la page est gérée à l'aide des deux feuilles de styles « index.css » et « default.css ».

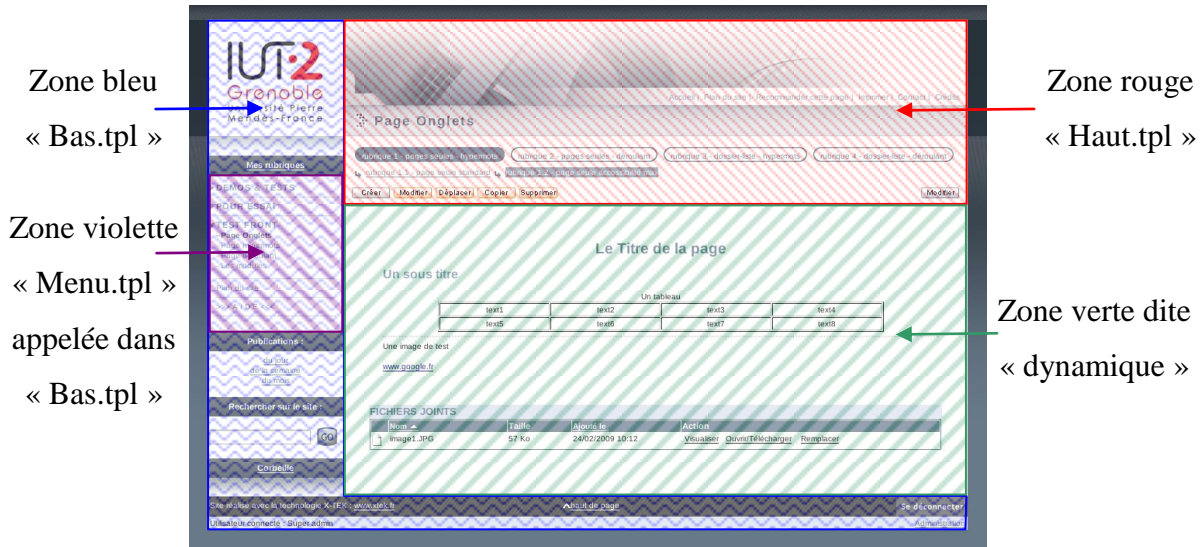


Figure 25: Différents templates de X-TEK

### 4.1.2. Code source

XTEK est une application constituée de 3147 fichiers PHP représentant 654 000 lignes de code. La Figure 26 montre l'arborescence des dossiers du code source de X-TEK.

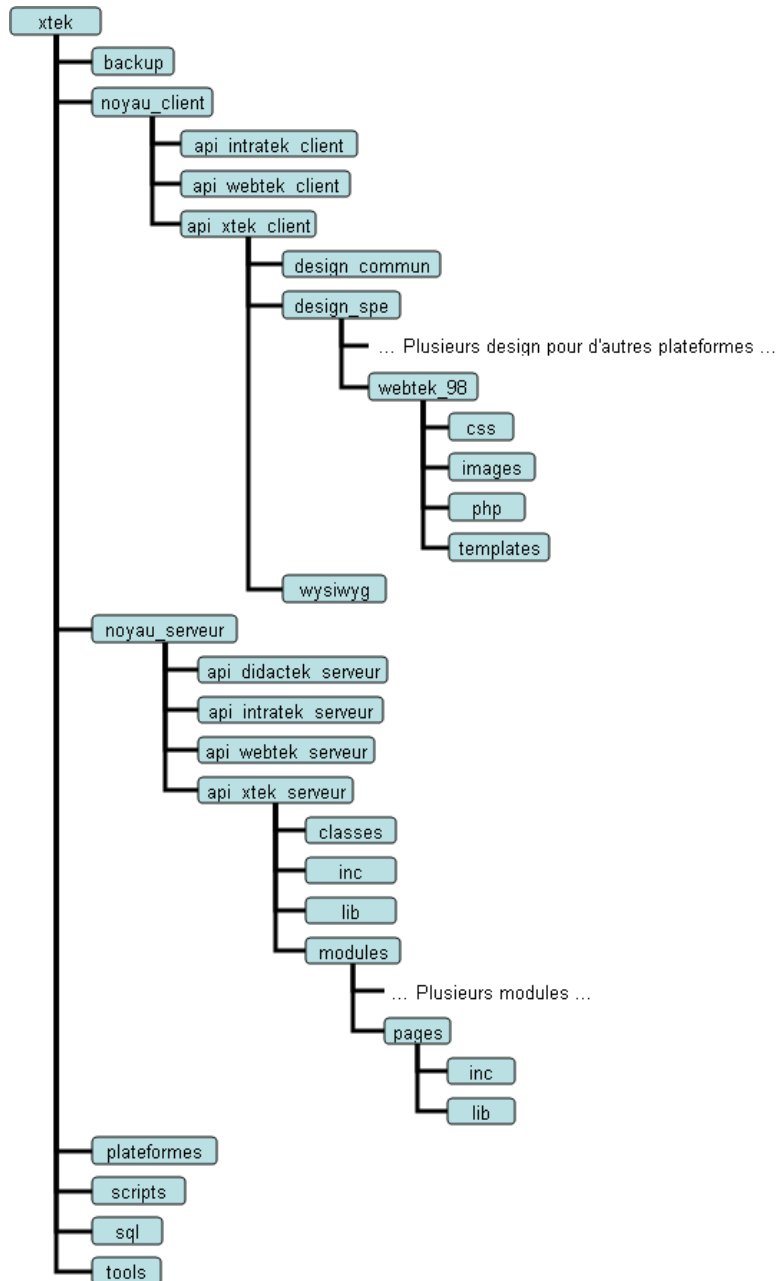


Figure 26 : Arborescence des dossiers du code source de X-TEK

Au cours de la mise en conformité nous travaillerons essentiellement sur deux grandes parties que sont le « noyau client » et le « noyau serveur », représentant plus des deux tiers du code.



Le « noyau client » gère la partie graphique de l'application et le « noyau serveur » est le cœur du fonctionnement de l'application.

Au sein du noyau client, nous retrouvons deux sous-parties :

- le « design commun » regroupant tous les éléments communs à l'ensemble des plateformes. Les éléments les plus importants pour notre tâche sont le fichier « general.js » contenant la majorité des fonctions javascript utilisées par les pages générées et le fichier « haut.inc.php » qui a pour rôle principal d'initialiser les variables utilisées par les templates des différentes plateformes.
- Le « design spécifique » regroupant les différentes chartes graphiques propres à chaque plateforme. Nous nous servons uniquement de la charte graphique du webtek-98 dans lequel se trouvent les templates, les feuilles de styles et les images propres à notre plateforme.

Le noyau serveur, quant à lui, représente la partie commune à l'ensemble de plateforme. Il regroupe toutes les bibliothèques et modules de l'application. Nous interviendrons essentiellement sur le module « pages » qui correspond concrètement aux modules « dossier une seule page », « dossier menu », « dossier liste », « formulaire en ligne » et ouvre indirectement accès au module « calendrier ». Il est à retenir deux fichiers importants au niveau des bibliothèques :

- « general\_design.inc.php » qui regroupe la majorité des fonctions appelées par « haut.inc.php » du « design commun » afin d'initialiser les variables utilisées par les templates. C'est dans ce fichier que nous retrouvons les différentes fonctions permettant de créer les rubriques et sous-rubriques, ainsi que les différents boutons d'administration et bien d'autres fonctionnalités.
- « switch\_section.inc.php » gère, quant à lui, les différents modules. C'est dans ce fichier que l'application appelle le module correspondant à la page demandée.

### **4.1.3. Patron de conception**

Les patrons de conception (Design pattern en anglais) sont des modèles de solution standard permettant de résoudre des problèmes récurrents d'architecture et de conception des logiciels.

Le patron de conception s'appliquant aux interfaces homme-machine est le MVC<sup>26</sup>. Le MVC est constitué de trois parties :

- Le modèle représente le comportement de l'application : traitements des données, interactions avec la base de données, etc.
- La vue correspond à l'interface avec laquelle l'utilisateur interagit.
- Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser.

Ce patron de conception permet de séparer le « Modèle », la « Vue » et le « Contrôleur », les rendant les plus indépendants possible. Ainsi toute modification d'une de ces parties n'aura peu voire aucune conséquence sur les autres. La Figure 27 montre l'architecture du patron de conception MVC.

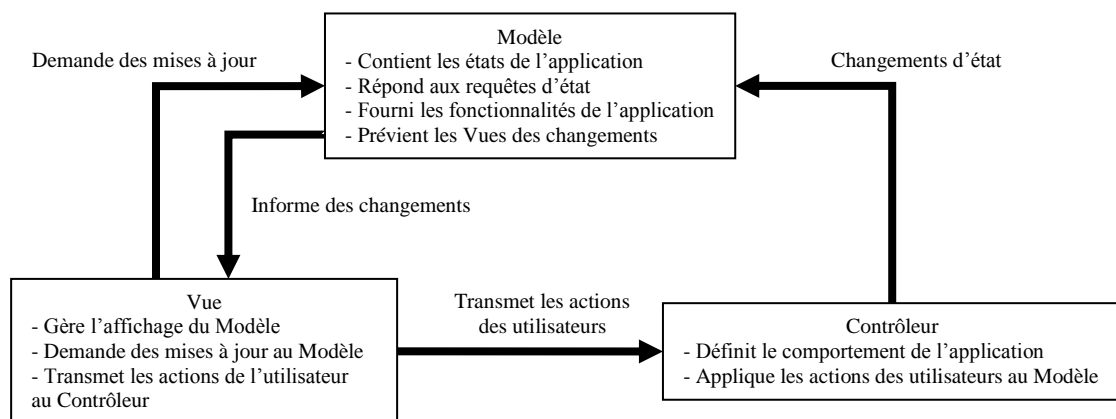


Figure 27 : Patron de conception MVC

Au niveau de la charte graphique, le patron de conception MVC était utilisé. Son utilisation a grandement facilité la mise en conformité de la charte graphique en séparant proprement la « Vue » du « Modèle » et du « Contrôleur ». En ce qui concerne les modules utilisés dans X-TEK, le patron de conception MVC a été peu voire pas du tout utilisé. Du code HTML et même des requêtes en base de données étaient mélangés dans le code PHP. Le manque de maturité de ces parties de code source a rendu difficile la mise en conformité des modules.

<sup>26</sup> MVC : Modèle Vue Contrôleur

## 4.2. Déroulement de la mise en conformité

La mise en conformité s'est déroulée de manière itérative, en commençant par la mise en conformité de la page d'accueil, afin de rendre la charte graphique accessible. Puis, nous avons élargi la mise en conformité à différentes pages simples, aux pages contenues dans les « dossier menu » et « dossier liste », pour terminer par les pages de consultation des formulaires.

Nous avons débuté la mise en conformité en nous appuyant sur le RGAA 1.0. Puisque nous avons connu une évolution du RGAA au cours de notre mise en conformité, nous avons alors repris entièrement ce qui a été fait en appliquant le RGAA 2.2. Faute d'outil d'analyse de l'accessibilité selon le WCAG 2.0, nous avons entrepris une analyse manuelle systématique des pages, en nous appuyant sur les 187 tests du RGAA 2.2 permettant de valider les critères de succès du WCAG 2.0 (Cf. annexe 2).

Comme il nous l'avait été demandé, nous avons autant que possible essayé de respecter l'apparence de la charte graphique précédant la mise en conformité, ainsi que le fonctionnement global de l'application.

Tout au long de cette mise en conformité, nous nous sommes formés en nous appuyant sur les ressources disponibles sur les sites traitant de l'accessibilité et du respect des standards [ALSAC1] [OPENW1].

## 4.3. Application des tests

Le RGAA 2.2 propose d'offrir une application concrète du WCAG 2.0. Pour ce faire, il décompose les critères d'accessibilité du WCAG 2.0 en différents tests. Cependant selon les critères d'accessibilité, ces tests s'étendent sur une trop grande plage de domaines techniques. Le RGAA les a alors regroupés en thématiques. Nous avons alors déroulé ces tests selon ces thématiques.

Pour chaque test, le RGAA nous renvoie vers les techniques du WCAG 2.0 correspondantes. [TWCAG2] Ces techniques se décomposent ainsi :

- Les techniques générales proposent des solutions et des comportements à atteindre sans reposer sur une technologie spécifique. Elles sont référencées par la lettre G suivie d'un numéro, par exemple G1 ou G119.
- Les techniques HTML et XHTML qui offrent des solutions d'accessibilité en HTML et XHTML. Elles sont référencées par la lettre H suivie d'un numéro comme H2 ou H12.
- Les techniques CSS qui offrent des solutions d'accessibilité aux feuilles de style. Elles sont référencées par la lettre C suivie d'un numéro comme C6 ou C30.
- Les techniques de script côté client qui offrent des solutions d'accessibilité pour les scripts exécutés sur l'agent utilisateur. Elles sont référencées par les lettres SCR suivies d'un numéro comme SCR1 ou SCR37.
- Les techniques de script côté serveur qui offrent des solutions d'accessibilité pour les scripts exécutés sur le serveur. Elles sont référencées par les lettres SVR suivies d'un numéro comme SVR1 ou SVR4.
- Les techniques SMIL<sup>27</sup> qui offrent des solutions d'accessibilité pour les contenus multimédia intégrés à l'aide du langage SMIL. Elles sont référencées par les lettres SM suivies d'un numéro comme SM1 ou SM14.
- Les techniques Texte qui proposent des solutions d'accessibilité pour les documents textes. Elles sont référencées par la lettre T suivie d'un numéro comme T1.
- Les techniques ARIA<sup>28</sup> qui offrent des solutions d'accessibilités afin d'augmenter l'accessibilité des applications Internet riches. Elles sont référencées par les lettres ARIA suivies d'un numéro comme ARIA1.
- Les échecs qui recensent les erreurs invalidant les critères d'accessibilités. Ils sont référencés par la lettre F suivie d'un numéro comme F1 ou F89.

---

<sup>27</sup> SMIL : Synchronized Multimedia Integration Language est un langage permettant l'intégration de contenus multimédias diversifiés en les synchronisant afin de permettre la création de présentations multimédias

<sup>28</sup> ARIA : Accessible Rich Internet Applications

### 4.3.1. Thématique Standard

Cette thématique vise à garantir le respect des standards du Web. Etre conforme aux standards est un gage d'interprétation correcte de nos pages par les différents agents utilisateurs.

Il existe 8 tests dans cette thématique qui permettent de vérifier :

- La présence d'une DTD positionnée avant la balise ouvrante HTML et déclarée conformément aux syntaxes validées par le W3C.
- La validité du code HTML / XHTML au regard de la DTD déclarée et l'absence de composants obsolètes.
- La présence d'un titre de page et d'une langue de traitement pour celle-ci, ainsi que la pertinence du titre vis-à-vis du contenu de la page.

Pour notre plateforme, la DTD était bien déclarée en tant que XHTML 1.0 Transitional et un titre était présent. Cependant le code de la page n'était pas conforme à la DTD déclarée, des composants obsolètes étaient présents et le titre de la page était le même pour toutes les pages.

Nous avons assez rapidement corrigé toutes les erreurs de validation de code en nous aidant du validateur de code XHTML en ligne du W3C « Markup Validation Service ». Le problème du titre de la page a, quant à lui, été résolu en concaténant le nom de la plateforme avec le nom de la page disponible dans le code source.

L'utilisation d'un outil de validation en ligne était problématique. Nos pages n'étant pas accessibles par Internet, nous étions obligés de copier le code source de chaque page générée pour pouvoir le valider. Ce fonctionnement était assez fastidieux mais nous avons pu accélérer la validation grâce à l'outil « Web Developer », disponible en plugins Firefox, qui, contrairement aux autres outils, permet de faire valider par « Markup Validation Service » le code source local, au lieu de simplement transmettre l'url de la page à valider au service W3C qui n'aurait pas pu y accéder. De plus, « Web Developer » offre des raccourcis clavier permettant d'optimiser encore la vitesse de traitement.

Les principales erreurs rencontrées sont décrites ci-dessous.

### 1) **Non respect des règles de syntaxe XHTML**

En XHTML, toutes les balises et tous les attributs doivent être écrits en minuscule. Les valeurs d'attributs doivent être encadrées entre guillemets et il est interdit d'écrire les attributs sous forme simplifiée. Ainsi, le code suivant n'est pas valide car la balise et les attributs sont en majuscule, la valeur de l'attribut « size » n'est pas entre guillemets et l'attribut « checked » est écrit sous forme simplifiée :

```
<INPUT TYPE= "CHECKBOX" SIZE=10 CHECKED/>
```

L'écriture correcte est la suivante :

```
<input type="checkbox" size="10" checked="checked"/>
```

Nous retrouvons aussi des oublis de fermeture de balise, en particulier pour les balises <br> et <hr> qui ne sont jamais fermées. Nous transformons alors ces balises en balises auto-fermantes <br/> et <hr/>.

### 2) **Absence d'attributs obligatoires**

En XHTML, il existe des attributs obligatoires pour certains éléments. Par exemple, pour les images « img », les attributs obligatoires sont la source « src » et l'alternative textuelle « alt » qui est généralement omise. De même pour les scripts « script » et les styles « style », l'attribut obligatoire est le type « type ».

### 3) **Usage des esperluettes**

L'esperluette en HTML est un caractère réservé qui marque le début d'une entité de référence de caractère, c'est-à-dire des caractères spéciaux, difficilement productible au clavier ou non représentable dans le jeu de caractères du document.

Nous la retrouvons, couramment utilisée dans les liens, afin de passer des paramètres qui seront récupérés par le serveur Web. Par exemple, il est fréquent de rencontrer des liens de ce type : <http://www.exemple.com/index.php?nom=durand&prenom=paul>

L'esperluette introduit ici le paramètre « prénom » ayant pour valeur « paul ». Or, l'usage de l'esperluette est réservé aux entités. Il faut donc représenter l'esperluette avec sa référence qui est « & ». Un lien correct devient ainsi dans le code source de la page : <http://www.exemple.com/index.php?nom=durand&prenom=paul>

Ces erreurs se retrouvent aussi pour tous les caractères spéciaux comme :

- Les signes < ou > qui, en HTML, délimitent les balises. Leurs représentations correctes sont « &lt; » et « &gt; ».
- Les guillemets anglais " qui, en HTML, délimitent les attributs que l'on doit représenter ainsi « &quot; ».
- Les caractères accentués en français comme « é, è, à, ç » que l'on représente ainsi « &eacute; , &egrave; , &agrave; &ccedil; ».
- Les symboles tels que le signe © (copyright) ou ® (registered trade mark) représentés par « &copy; » et « &reg ; ».

#### 4) Mauvaise imbrication des éléments

En HTML, le chevauchement des éléments est interdit. Ainsi le code suivant n'est pas valide :

```
<a><strong>lien</a></strong>
```

Bien que peut fréquent, ce type d'erreur se retrouvait au niveau des contenus complexes nécessitant un important traitement dans le code PHP avec plusieurs niveaux de structures conditionnelles. Généralement, l'erreur provient de l'ouverture (ou de la fermeture) d'un élément HTML alors que sa fermeture (ou ouverture) est effectuée dans une structure conditionnelle plus ou moins complexe. Ceci implique plusieurs fermetures (ou ouvertures) différentes selon la complexité de la structure conditionnelle et augmente, de par le fait de la

non unicité de la fermeture (ou ouverture), les risques de mauvaises imbrications avec les éléments environnants.

### 5) **Ecriture de code HTML dans un élément « script »**

En XHTML, les éléments « script » sont déclarés comme ayant pour contenu du #PCDATA. Il en résulte que tout caractère « < » sera considéré comme le début d'une balise et toute esperluette comme le début d'une entité. La solution préconisée est d'insérer le code javascript dans une section CDATA afin de rendre ce contenu non interprétable comme le montre le code suivant :

```
<script type="text/javascript">
  <![CDATA[
    *le code javascript/*
  ]]>
</script>
```

Or, les sections CDATA sont mal interprétées par les anciens navigateurs qui analysent le code en HTML et non en XML. Nous nous retrouvons donc dans le même cas qu'au début des balises « script ».

Historiquement, au début de l'usage des balises « script », il était recommandé d'encadrer par des balises de commentaires, le code source du script, afin de garantir la compatibilité avec les anciens navigateurs de l'époque qui ne savaient pas interpréter les balises « script » et qui affichaient alors directement le code source. Aujourd'hui, la solution est de commenter le début et la fin de section afin que les navigateurs ne supportant pas les sections ne les interprètent pas. Le codage de l'élément « script » devient donc ceci :

```
<script type="text/javascript">
  //< ![CDATA[
    *le code javascript/*
  //]]>
</script>
```



Il devient ainsi évident qu'il ne faut pas utiliser la chaîne de caractères « ]]> » dans le code source de notre script, sous peine de sortir inopinément de notre section CDATA.

### **4.3.2. Thématique « Navigation »**

Cette thématique vise à garantir une navigation aisée à travers la page et dans les liens vers d'autres documents ou pages.

Il existe 36 tests que nous pouvons regrouper en 7 sous-ensembles de cette thématique.

#### **1) Accessibilité des liens doublant les zones cliquables côté serveur**

Afin de rendre accessibles les « zones cliquables » côté serveur comme les régions sélectionnables d'une image utilisant l'attribut « ismap », il nous est demandé de doubler les « zones cliquables » en fournissant, immédiatement après l'image, des liens remplissant le même rôle.

Au niveau de X-TEK, il n'existe pas de cas concret d'image exploitant les « zones cliquables » côté serveur. Toutefois, puisqu'il est possible pour un contributeur d'importer du code HTML, il nous faut tout de même évoquer leur possible utilisation dans X-TEK. La seule solution est de rappeler aux contributeurs de penser à doubler les « zones cliquables » avec des liens.

#### **2) Identification du rôle des liens**

Selon le handicap de l'utilisateur ou son mode de consultation du site Web, il est possible qu'il ne puisse pas saisir parfaitement le rôle des différents liens qui lui sont présentés. Une personne souffrant d'un déficit visuel peut ne pas percevoir le site comme un utilisateur ayant une bonne vision. Son approche sera séquentielle et il n'aura pas accès au contexte visuel.

Par exemple, un site Web propose le téléchargement d'un logiciel. Il affiche en début de page, le nom du logiciel suivi d'une description et enfin, le lien permettant le téléchargement intitulé simplement « télécharger ». Un utilisateur sans déficience aura visuellement accès au nom du logiciel et au lien téléchargé, et il comprendra aisément que ce lien permet de télécharger le logiciel. Alors qu'un utilisateur accédant séquentiellement à la page peut ne

plus pouvoir associer ce lien de téléchargement avec le logiciel car il aura été obligé de lire tout le contenu de la description avant de parvenir au lien.

Le RGAA demande que tout lien possède un intitulé. Cet intitulé peut être soit le contenu textuel du lien ou bien l'alternative textuelle d'un élément non textuel contenu dans le lien. Il faut que l'on puisse identifier clairement l'action ou la destination des liens ou des boutons à partir de leurs intitulés ou de leurs intitulés additionnés aux contenus environnants. Il ne faut pas que des liens ou boutons possédant des intitulés identiques provoquent des actions différentes. De plus, il faut que tout lien permettant le téléchargement d'un document précise le format, la langue et la taille de ce document. Pour ce faire, le RGAA nous renvoie vers les techniques G91, G197, H30, H33 et F67.

Nous avons alors revu chaque lien et bouton individuellement, afin d'isoler leur fonction et leur donner un intitulé correspondant à leur rôle. Nous avons regroupé chaque lien ayant les mêmes fonctions afin de leur donner des intitulés identiques. Lorsque c'était possible, nous avons fait en sorte que l'intitulé du lien ne soit composé que du contenu textuel du lien. Dans les autres cas, nous avons exploité le plus possible les alternatives textuelles et les attributs « title » des liens.

Pour ce qui est des liens permettant le téléchargement de documents, la tâche est un peu plus compliquée. En effet, il nous est impossible de déterminer si un lien fourni par un contributeur permet de télécharger un document. Nous ne pouvons contrôler ce type de lien que s'ils sont générés après une procédure établie permettant de charger le fichier en question sur le serveur et de récupérer les informations nécessaires auprès du contributeur. Ce type de procédure existe déjà dans les dossiers une seule page. Nous nous sommes donc intéressés uniquement à cette partie et nous ne pouvons que laisser les contributeurs veiller à bien renseigner leurs liens pour le téléchargement de documents en dehors de la plateforme X-TEK.

Au niveau du dossier une seule page, X-TEK possédait déjà une gestion de la taille du document et une gestion du format, mais pas de gestion de la langue. La taille est obtenue aisément à partir du fichier. Le format est, quant à lui, obtenu à partir de l'extension du fichier et ceci peut poser des problèmes. En effet, un fichier ne présentant pas une extension connue ou tout simplement erronée, sera considéré comme un document binaire ou au pire comme un

fichier d'un autre format. De plus, il serait judicieux de ne pas se reposer uniquement sur l'extension du fichier, afin de restreindre le chargement de fichiers indésirables comme des exécutable et éviter ainsi le détournement de notre plateforme vers des usages non désirés.

Pour pallier ce problème, nous avons mis en place une détection du format à l'aide de la librairie PHP « magic mime ». Les « magic mime » permettent de détecter le format en recherchant des octets ou suites d'octets particuliers à l'intérieur d'un fichier. Le module PEAR de PHP possédant la fonction de recherche à l'aide des « magic mime », nous avons activé cette fonctionnalité et modifié X-TEK afin qu'il puisse les utiliser à la place des extensions. De même, X-TEK utilise une correspondance statique entre l'extension et le type mime des fichiers dans un tableau, nous avons alors modifié la table « fichier » au niveau de la base de données de X-TEK afin d'associer le type mime au fichier. X-TEK peut maintenant reconnaître une multitude de formats de fichiers sans se baser uniquement sur l'extension.

Pour la gestion des langues, nous avons créé une table langue, en base de données, faisant correspondre au code de langue ISO 639-1 le nom de la langue. Nous avons alors modifié la table « fichier » afin de rattacher un code langue au fichier, ainsi que modifié le formulaire de création et de modification pour permettre la sélection du code langue à associer au fichier. Pour des raisons de mise en conformité des différentes plateformes X-TEK, nous n'avons pas créé de clé étrangère entre le code langue de la table fichier et le code langue de la table langue.

Finalement, nous avons pu modifier les liens permettant le téléchargement des fichiers, en leur rajoutant les informations de format, de taille et de langue.

### **3) Regroupement des liens importants et présence de contournement ou d'accès rapide aux groupes de liens et au contenu**

Le regroupement des liens permet de donner de la cohérence aux liens. On les utilise pour les barres d'outils et les menus, regroupant logiquement les liens ensemble et facilitant ainsi la compréhension du rôle de ces liens pour les utilisateurs ayant un accès séquentiel à la page. La technique H50 propose des solutions de regroupement à l'aide d'élément « map » ou bien sous forme de liste ordonnée (élément « ol ») ou non ordonnée (élément « ul »). Nous avons fait le choix des listes non ordonnées pour nos groupements de liens car la distinction des liens est plus facile sous forme de liste que dans un élément « map ». En effet, un groupement

dans un élément « map » nécessitera de séparer les liens visuellement à l'aide de caractères et la distinction serait moins marquée avec une synthèse vocale qui n'aurait pas d'éléments structurels pour marquer clairement le passage d'un lien à un autre. Le choix des listes non ordonnées au lieu des listes ordonnées, vient du fait que ne possédant pas de numérotation, les listes non ordonnées sont plus légères lors d'une retranscription vocale.

De plus, les techniques G1, G123 et G124 préconisent de fournir des moyens d'accès rapides et de contournement aux différents groupements de liens mais aussi au contenu de la page. Ceci est particulièrement utile pour les utilisateurs accédant séquentiellement à la page, leur permettant de passer directement au contenu de la page et aussi de passer rapidement les blocs de contenu ne les intéressant pas.

Pour ce faire, le RGAA propose de rendre les blocs d'information identifiables en leur fournissant des ancres soit en identifiant le bloc d'information lui-même en utilisant un attribut « ID » ou bien en faisant appel à un élément « a » positionné avant le bloc de contenu ou bien en tant que premier enfant de l'élément servant de conteneur du bloc d'information. Nous avons choisi de placer arbitrairement des ancres avant les blocs de contenus car il nous paraissait simplement plus logique qu'un lien d'accès rapide à un bloc pointe juste avant le bloc d'information. Nous avons par la suite fourni un lien à l'intérieur du bloc pointant sur une ancre suivant directement le bloc d'information. Nous avons en effet choisi de ne pas pointer directement sur le bloc d'information suivant afin de garder une parfaite indépendance entre les blocs.

Pour finir, nous avons fourni une barre d'accès rapide accessible, dès le début de la page, permettant d'atteindre rapidement les principaux blocs d'information.

#### **4) Présence et l'usage des outils de navigation**

Le RGAA demande la présence de différents outils de navigation :

- Des menus ou des barres de navigation en nous renvoyant vers la technique H59 qui nous propose une solution permettant de naviguer dans un groupe de pages à l'aide de l'élément « link ». Mais cette technique ne trouve pas d'application dans le fonctionnement actuel de X-TEK.

- Un plan du site en nous renvoyant vers les techniques G63 et G64. Il nous faut veiller à la cohérence du plan du site par rapport à l'architecture réelle du site. Il faut aussi que le plan soit accessible à partir de toute page du site et en particulier à partir de la page d'accueil.
- Un moteur de recherche en nous renvoyant vers la technique G161.
- Un fil d'Ariane en nous renvoyant vers la technique G65.

X-TEK ne possédait pas de fil d'Ariane. Nous avons donc créé un fil d'Ariane se basant uniquement sur une logique arborescente que nous avons due intégrer dans la charte graphique. Nous avons poussé la logique du fil d'Ariane jusqu'à l'intérieur des dossiers listes permettant ainsi une accessibilité même dans ces sous-structures que sont les dossiers liste.

De plus, il nous faut fournir une indication de position dans les menus permettant aux utilisateurs de facilement se repérer (technique G128). Nous avons opté pour l'usage de l'élément « strong » pour englober le lien de la position courante dans les barres de navigation. L'utilisation de cet élément est un des moyens les plus logiques permettant de faire ressortir l'importance du contenu encadré en donnant un effet de « graisse » sur le contenu. Il est d'autant plus adéquat qu'il est accessible même aux non-voyants puisque étant un élément et non simplement une mise en forme graphique, il sera interprété par les agents utilisateurs tels que les revues d'écran.

#### **5) Présence d'avertissement préalable à toutes ouverture de nouvelle fenêtre**

Afin de ne pas désorienter les utilisateurs, il faut prévenir de toute ouverture de fenêtre dans les liens, les formulaires ou dans l'usage de script. Nous avons enrichi les intitulés et/ou les attributs « title » de tous les liens et éléments de formulaire le nécessitant afin d'informer au préalable de l'ouverture d'une nouvelle fenêtre. Nous avons aussi renseigné les attributs « target » afin de fournir un nom adéquat aux nouvelles fenêtres créées.

#### **6) Absence d'action automatique sans action de l'utilisateur**

Dans la même logique que le point précédant, il faut bannir toute action automatique des pages pour ne pas déstabiliser les utilisateurs à moins de fournir un moyen d'arrêter les

automatismes. Ceci englobe les rafraîchissements automatiques et les redirections non sollicitées. Nous avons vérifié qu'aucune action automatique n'était présente.

#### **7) Absence de piège de navigation ainsi que cohérence de la navigation**

Il est important de donner les mêmes capacités de navigation à tous les utilisateurs et bannir tout ce qui pourrait nuire à leur navigation dans la page. Il faut donc veiller à la cohérence des éléments de navigation en conservant d'une page à l'autre le même ordre d'apparition, la même apparence graphique et logique, la position identique des liens de contournement et d'accès rapide.

En nous aidant des techniques G59, G61 et F66, nous avons vérifié à travers le fonctionnement interne de X-TEK que rien ne vient bouleverser la cohérence des éléments de navigation.

Il nous faut aussi éviter les pièges de navigation comme altérer le déplacement entre éléments à l'aide du clavier. Il faut aussi faire en sorte que la navigation au clavier suive un ordre logique par rapport au contenu.

Nous avons vérifié que tout élément pouvant recevoir le « focus » dans la page était atteignable au clavier et qu'il était possible d'aller aussi bien à l'élément précédent qu'à l'élément suivant. L'ordre de déplacement au clavier reste logique par rapport au contenu.

Une observation est à faire concernant cette partie. Nous avons eu plusieurs allers-retours entre cette partie et la thématique présentation concernant la structure même de la page. Nous en reparlerons au niveau de cette thématique que nous aborderons par la suite.

#### **4.3.3. Thématique « Scripts »**

Cette thématique vise à garantir l'accessibilité des scripts ainsi que des contenus et des éléments qu'ils créent ou modifient.

Il existe 13 tests que nous pouvons regrouper en 5 sous-ensembles de cette thématique.

## **1) Accessibilité des contenus dynamiques ou modifiés dynamiquement**

Il faut que tout contenu modifié dynamiquement reste accessible. Par exemple, dans le cas de l'ajout ou de la modification d'une image, il faut que l'alternative textuelle de l'image soit aussi modifiée.

Il faut que les contenus dynamiques modifiés ou ajoutés soient positionnés immédiatement après l'élément ayant entraîné leur ajout ou modification, afin de conserver l'ordre d'accès au clavier.

De plus, il faut que tout contenu dynamique utile à la compréhension ou l'exploitation de la page soit accessible sans scripts. Il nous faut fournir une alternative aux contenus des scripts.

Au niveau de X-TEK, les scripts ne modifiaient pas dynamiquement le contenu c'est-à-dire sans rechargement de la page. Les seules modifications dynamiques sont des modifications d'attributs « class » dans certains éléments des menus permettant de masquer certaines parties. Nous avons juste veillé à obtenir une équivalence de contenu avec ou sans script. Ainsi les sous-menus repliés à l'aide de javascript sont dépliés lors d'une consultation sans script.

## **2) Accessibilité des scripts au clavier**

Il faut que tout gestionnaire d'événements javascript rattaché à un élément permettant d'exécuter un script soit accessible au clavier. Prenons par exemple un bouton créé à l'aide d'un lien image avec une image qui change lorsque l'on passe la souris au-dessus. Cet effet est obtenu en utilisant le gestionnaire d'évènements « onmouseover » sur l'élément « a » de notre lien image. Lors d'une navigation au clavier, l'image ne changera pas d'apparence car le gestionnaire d'évènements « onmouseover » ne fonctionne qu'avec l'interaction de la souris. Il nous faut alors doubler notre gestionnaire d'évènements « onmouseover » avec le gestionnaire d'évènements « onfocus » pour fournir la même fonctionnalité.

Dans X-TEK, nous avons rarement eu besoin de modifier les fonctionnements des gestionnaires d'évènements. Cependant nous avons tout de même dû mettre en place des gestionnaires d'évènements « onclick » sur tous les liens utilisant des scripts pour ouvrir des fenêtres réduites ou réaliser des fonctions complexes. Prenons, par exemple, le lien permettant

d'ouvrir le formulaire de contact qui s'ouvre à l'aide de la fonction javascript « EnvoiSuggestion ». Le code source était le suivant :

```
<a href="EnvoiSuggestion()" ... > ... </a>
```

Sous cette forme, la désactivation des scripts rendait la fonctionnalité contact inaccessible. Nous avons alors modifié le lien afin de rendre la fonctionnalité contact accessible sans script en passant l'appel de la fonction javascript dans le gestionnaire d'évènements « onclick » et en fournissant en tant que « href » le lien effectuant la même fonction. De plus, nous avons aussi fait en sorte de conserver une ouverture dans une nouvelle fenêtre en utilisant l'attribut « target » pour rester cohérent avec l'intitulé du lien informant de l'ouverture de la page dans une nouvelle fenêtre. Le code source de la fonctionnalité devient ainsi :

```
<a target="SUGGESTION" href="index.php ?special=..." onclick="EnvoiSuggestion() ;return false ;" ...> ... </a>
```

Il est à noter deux points importants :

- La présence de « return false ; » qui permet d'interdire au navigateur de suivre la destination du lien lorsque l'évènement « onclick » est déclenché permettant ainsi de ne pas ouvrir la fenêtre sans script en même temps que la fenêtre script.
- La seule utilisation du gestionnaire d'évènements « onclick » qui permet aussi bien le déclenchement de l'évènement à la souris qu'au clavier.

### **3) Contrôle de la navigation par l'utilisateur**

Il faut permettre à l'utilisateur de pouvoir bloquer l'apparition d'une fenêtre d'alerte non sollicitée ainsi que toute mise à jour automatique du contenu. Mais aussi, il faut interdire tout changement de contexte, suite à une action de l'utilisateur sans validation de sa part ou information préalable.

X-TEK n'étant pas sujet à l'apparition intempestive de fenêtres d'alertes, à la modification du contenu dynamiquement et au changement de contexte comme l'ouverture d'applications externes, nous avons juste veillé à bien prévenir l'utilisateur avant toute ouverture de fenêtre, ce qui a été fait dans la thématique Navigation.



#### **4) Respect du rôle des éléments et absence de suppression du focus**

Il est interdit d'utiliser des éléments non prévus pour la navigation en tant que lien à l'aide de script. Prenons l'exemple d'une simple image non contenue dans un élément lien. Si nous lui ajoutons un gestionnaire d'évènements « onclick » qui déclenche un script permettant d'ouvrir une nouvelle page, notre image se comportera comme un lien. Cependant, ce lien image ne sera pas accessible au clavier puisqu'une image ne peut obtenir de focus.

De plus, il ne faut pas faire en sorte, à l'aide de script, de supprimer le focus sur un élément pouvant le recevoir. Cet élément serait uniquement accessible à la souris et non au clavier.

Au niveau de X-TEK, nous n'avons pas rencontré d'élément utilisé comme élément de navigation, ni de script supprimant le focus.

#### **5) Absence de limite de temps**

Il faut éviter de limiter en temps d'interaction avec les interfaces comme des formulaires de saisie à moins que cela ne soit impérativement nécessaire comme dans les cas exceptionnels où les données seraient susceptibles de changer dans un laps de temps très court (par exemple formulaire d'achats aux enchères). Dans le cas contraire, des utilisateurs à mobilité réduite ou ayant des déficiences intellectuelles seraient dans l'incapacité d'exploiter ces interfaces dans les temps impartis.

Il faut aussi faire en sorte de conserver les informations de session afin d'éviter la perte d'information lors de l'expiration d'une session authentifiée. Ce mécanisme permet aux utilisateurs ayant besoin de plus de temps que les autres de ne pas perdre les informations qu'ils auraient déjà saisies lorsque leur session se verrait fermée par le serveur pour cause d'inactivité ou de dépassement de délai.

Au niveau de X-TEK, nous ne sommes pas soumis à ces problèmes techniques. Nous avons alors simplement vérifié qu'aucun problème de ce genre n'était présent.

### **4.3.4. Thématique « Images »**

Cette thématique vise à garantir l'accessibilité des images en se basant essentiellement sur les alternatives textuelles des images.

Il existe 11 tests que nous pouvons regrouper en 3 sous-ensembles de cette thématique.

### **1) Alternatives textuelles et descriptions longues**

Toute image doit posséder une alternative textuelle fournie par l'attribut « alt ». Cette alternative peut être vide dans le cas d'image uniquement décorative. Dans tous les autres cas, elle doit impérativement être pertinente dans le contexte de l'image et surtout la plus concise possible.

Pour certaines images transportant plus d'informations, il est nécessaire de fournir une description longue retranscrivant les informations de l'image. Cette description longue peut, soit être contenue dans la même page et être accessible textuellement après l'image ou atteignable à l'aide d'un lien suivant l'image. Elle peut aussi être disponible sur une autre page et être rendue accessible à l'aide de l'attribut « longdesc » de l'image.

Au niveau de X-TEK, nous n'avons qu'un nombre limité d'images présentes. Nous avons aisément pu les mettre en conformité. Nous avons en plus réduit leur nombre en convertissant la plupart des menus utilisant des images, en images gérées par les feuilles de styles. De plus, nous n'avons aucune image nécessitant une description longue. Le respect de cette partie se fera donc essentiellement en formant les contributeurs pour qu'ils utilisent les fonctionnalités d'accessibilité de l'éditeur de texte enrichi afin de fournir une alternative textuelle à chaque image le nécessitant.

### **2) Accessibilité des CAPTCHA**

Rendre accessible un test CAPTCHA<sup>29</sup> n'est pas très aisé. Il faut fournir une description de l'action à accomplir pour valider le CAPTCHA. Par exemple, un CAPTCHA nécessitant de retranscrire le mot fourni par l'image aura pour alternative textuelle « Entrez le mot contenu dans l'image ». Si une alternative au CAPTCHA est présente, l'alternative doit donner les instructions à suivre permettant d'atteindre cette alternative.

X-TEK ne comporte pas de procédure de test à l'aide de CAPTCHA.

---

<sup>29</sup> CAPTCHA : Il s'agit d'une forme de test de Turing permettant de différencier, de manière automatisée, un utilisateur humain d'un ordinateur

### **3) Cohérence des alternatives textuelles**

Il faut que l'alternative textuelle d'une image soit cohérente au sein de la page et à travers les pages du site.

Sur la partie de X-TEK que nous traitons, nous avons vérifié la cohérence des alternatives textuelles des images sur la même page et à travers les autres pages du site. Il nous est cependant impossible de garantir cette cohérence sur les images importées par les contributeurs. En effet, bien qu'il existe une bibliothèque d'images au sein de X-TEK, il n'y a pas d'association conservée entre l'image et son alternative textuelle. Pour rendre la tâche des contributeurs plus facile, il faudrait mettre en place une bibliothèque d'images associant les attributs d'accessibilité aux images.

#### **4.3.5. Thématique « Couleurs »**

Cette thématique vise à garantir l'accessibilité des contenus en faisant en sorte d'avoir suffisamment de contraste entre les couleurs de texte et de fond, ainsi qu'en vérifiant l'accessibilité des contenus par d'autres moyens que la couleur.

Il existe 16 tests que nous pouvons regrouper en 2 sous-ensembles de cette thématique.

#### **1) Utilisation d'un autre moyen que la couleur pour convoier de l'information**

Il faut veiller à fournir un autre moyen que la couleur pour identifier des contenus auxquels on fait référence et, d'une manière générale, il est préférable de souligner l'importance d'un élément en utilisant des marqueurs sémantique comme l'élément « strong ».

Prenons pour exemple un formulaire qu'un utilisateur vient de saisir. L'utilisateur soumet son formulaire en ligne mais le serveur le refuse et lui soumet à nouveau son formulaire pré-rempli en l'informant de corriger les éléments colorisés en rouge. L'utilisateur non-voyant sera dans l'incapacité de discerner les champs nécessitant une correction.

Au niveau de X-TEK, il n'est pas fait référence à des éléments en se basant sur la couleur. Le respect de cette partie se fera encore au niveau de la formation des contributeurs.

## 2) **Contraste suffisant**

Il faut qu'il y ait un contraste suffisant entre la couleur du texte et la couleur de fond et ce, même pour les textes contenus dans des images.

Pour ce faire, le RGAA, en s'appuyant sur les techniques G17, G18, G145, G174 et F83, préconise pour les textes inférieurs à 150% ou 120% gras de la taille de texte par défaut, au minimum un ratio de contraste supérieur ou égal à 4,5 et au mieux supérieur ou égal à 7, et pour les textes supérieurs ou égaux à 150% ou 120% gras de la taille de texte par défaut, au minimum un ratio de contraste supérieur ou égal à 3 et au mieux supérieur ou égal à 4,5.

Le ratio de contraste est calculé à partir de la formule suivante :

$$Ratio = \frac{L1 + 0.05}{L2 + 0.05}$$

Où L1 correspond à la luminance de la couleur la plus claire entre la couleur de texte et la couleur de fond et L2 correspond à la luminance de la couleur la plus sombre.

La luminance se calcule à partir de la formule suivante :

$$L = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B$$

Où R est la couleur rouge, G est le couleur verte et B la couleur bleue ; couleurs définies à partir de :

$$\text{Si } X_{sRGB} \leq 0.03928 \text{ alors } X = X_{sRGB} / 12.92 \text{ sinon } X = ((X_{sRGB} + 0.055) / 1.055)^{2.4}$$

Où X remplace R, G ou B et  $X_{sRGB} = X_{8bit} / 255$ , et  $X_{8bit}$  est la valeur décimale de la couleur correspondante sur 8 bits.

Après recensement des couleurs utilisées dans X-TEK, nous arrivions à un total de 51 couleurs différentes. Le nombre de couleurs étant bien trop important, nous avons été obligé d'en fusionner certaines afin de pouvoir les mettre en conformité vis-à-vis des valeurs de contraste. Nous avons donc confronté chaque couleur de fond en effectuant un ratio entre la brillance de chaque couleur et les avons aussi confrontées les unes aux autres en estimant la différence de couleur. Puis nous avons renouvelé l'opération pour les couleurs de texte. Nous

avons alors pu mettre en évidence les couleurs les plus proches que nous avons fusionnées dans une couleur intermédiaire.

Après ce traitement, nous avons atteint un nombre de couleurs plus raisonnable de 35 et nous sommes enfin passés à la mise en conformité des couleurs de X-TEK.

Afin d'accélérer les traitements et traiter uniquement les couples de couleurs texte et fond posant problème, nous nous sommes aidés du plugin firefox « Colour Contrast Analyser » permettant de contrôler automatiquement les ratios de contraste pour chaque couple de couleur de texte et de fond présents sur la page courante. Nous avons pu assez facilement atteindre le contraste recommandé pour chaque couple de couleurs.

#### **4.3.6. Thématique « Présentation »**

Cette thématique veille au respect de la séparation entre la mise en forme et le contenu, ainsi qu'à garantir l'accessibilité des contenus mis en forme à l'aide de feuilles de styles aussi bien du point de vue visuel que logique.

Il existe 18 tests que nous pouvons regrouper en 6 sous-ensembles de cette thématique.

##### **1) Séparation entre la mise en forme et le contenu**

Le RGAA impose de séparer le contenu de sa mise en forme. Ainsi toute la présentation de la page doit passer en feuilles de styles et tout élément de présentation HTML ou tout usage d'élément à des fins de présentation devient interdit. Il faut aussi que, de leur côté, les feuilles de styles ne génèrent aucun contenu porteur d'information.

Prenons par exemple une image de fond appliquée par une feuille de style. Si cette image apporte de l'information par du texte sous forme d'image ou bien encore si elle représente un graphique, alors nous n'aurions aucun moyen de rendre ce contenu accessible à des non-voyants.

De plus, il faut autant que possible essayer de transformer tout élément non textuel par une mise en forme effectuée à l'aide de styles. Ceci permet d'améliorer grandement l'accessibilité de ces éléments.

Au niveau de X-TEK, nous avons passé tous les éléments de mise en forme en feuilles de styles. Il s'agissait essentiellement d'éléments « font », ainsi que d'attributs « color » et « background ». Nous n'avons pas rencontré d'élément utilisé à des fins de présentation. Cependant, nous avons eu à terminer le passage d'une structure de mise en page du contenu sous forme de tableaux vers une mise en forme à l'aide de styles. En effet, X-TEK était au départ uniquement structuré par une imbrication de tableaux. Lorsque X-TEK a été transformé afin d'utiliser des templates, une partie de la mise en page est passée en style comme les menus, le bandeau supérieur et le bas de page. Toutefois, le contenu n'avait pas évolué et son code avait simplement été injecté dans un élément « div ». Nous avons donc transformé ces tableaux en éléments « div » afin de dédier les tableaux uniquement aux données.

Parallèlement au regroupement des liens de la thématique « Navigation », nous avons remplacé les éléments non textuels des menus sous forme d'hypermots et d'onglets, ainsi que du menu d'administration.

Les menus hypermots étaient formés en utilisant une image en guise de puce suivie par le lien du menu. Un exemple de menu hypermot est donné en Figure 28.



**Figure 28: Exemple de menu Hypermot**

Le menu est devenu une liste à laquelle nous avons substitué, à la puce par défaut, l'image de la puce des hypermots.

Les menus onglets étaient formés en utilisant un tableau ayant dans chaque cellule un tableau formant un onglet. Le tableau de l'onglet était constitué d'une première cellule vide ayant pour image de fond l'ouverture de l'onglet, puis une cellule contenant le lien du menu ayant pour image de fond les bords supérieur et inférieur de l'onglet et enfin une cellule vide ayant pour image de fond la fermeture de l'onglet. Un exemple de menu onglet est donné en Figure 29.



**Figure 29 : Exemple de menu Onglet**

Le menu est encore une fois devenu une liste contenant 3 éléments « div » imbriqués permettant, à partir des feuilles de styles, d'effectuer la même procédure d'affichage c'est-à-dire : dans les deux premiers niveaux de « div » nous affichons les ouvertures et fermetures de l'onglet, puis dans le dernier « div » nous répétons les bordures supérieures et inférieures de l'onglet en fond et enfin nous avons le lien du menu.

Enfin dans le menu d'administration, nous avons un tableau possédant deux cellules. La cellule de gauche avait son contenu aligné à gauche et celle de droite avait son contenu aligné à droite. Les boutons des fonctions étaient obtenus à l'aide de liens image dont les images contenaient le texte des boutons. La Figure 30 montre un exemple de menu d'administration.



**Figure 30: Exemple de menu d'administration**

Nous avons modifié le menu en le transformant en deux listes distinctes. L'apparence des boutons est obtenue en affichant, à l'aide de styles, l'image dépourvue de texte en fond d'élément « a ». La première liste est rendue flottante à gauche et la seconde flottante à droite.

Toutefois, ce travail sur les menus se révélera au final parfaitement inutile à l'exception de celui sur le menu d'administration. En effet, la partie suivante nous obligera à changer la logique même de fonctionnement des menus de X-TEK.

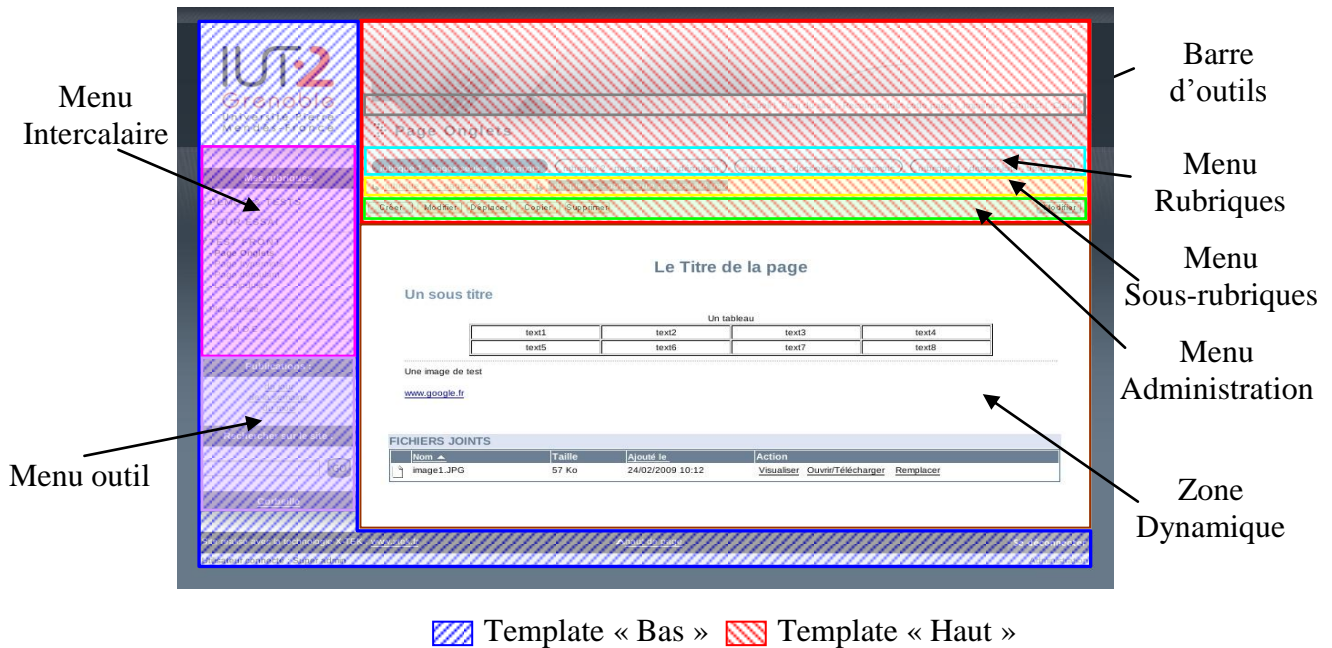
## 2) Ordonnement logique des contenus

Il nous est demandé de rendre l'ordre de restitution de l'information indépendante de sa mise en forme afin que tout contenu ayant une relation d'ordre logique avec un autre contenu apparaisse conformément à cet ordre.

Prenons par exemple trois blocs d'information A, B et C. S'ils sont parfaitement indépendants entre eux, alors nous pouvons les ordonner dans n'importe quel ordre. Par contre, si les blocs A et B possèdent une relation nous obligeant à lire A pour comprendre B, il faut alors présenter A avant B. Les séquences possibles deviennent alors A, B, C ou C, A, B.

Au niveau de X-TEK, nous avons rencontré une certaine difficulté avec les menus afin de respecter un ordre de séquence logique. La Figure 31 nous montre la structure de la page

avant modification ainsi que la disposition des templates et de leurs composants sur la page. Puisque le template « haut » apparaît avant le template « bas » dans le code source de la page, les menus des rubriques, sous-rubriques et d'administration apparaissent avant le menu intercalaire contenu dans le template « bas ». Ce qui implique pour les utilisateurs ayant un accès séquentiel à la page, l'apparition des rubriques et des sous-rubriques, puis du contenu de la page avant de pouvoir accéder au menu principal qu'est le menu intercalaire.



**Figure 31 : Structure de la page standard**

Un séquençement logique de la page aurait demandé d'avoir les menus regroupés sans que viennent s'intercaler un autre bloc d'information entre eux. Nous avons alors réussi, après avoir utilisé toutes les possibilités des feuilles de styles, à regrouper les menus tout en conservant un affichage identique. Nous avons alors obtenu un nouveau découpage de la page, comme présenté sur la Figure 32, dans lequel le menu intercalaire est intégré dans le template « haut » et précède les menus rubrique et sous-rubrique.



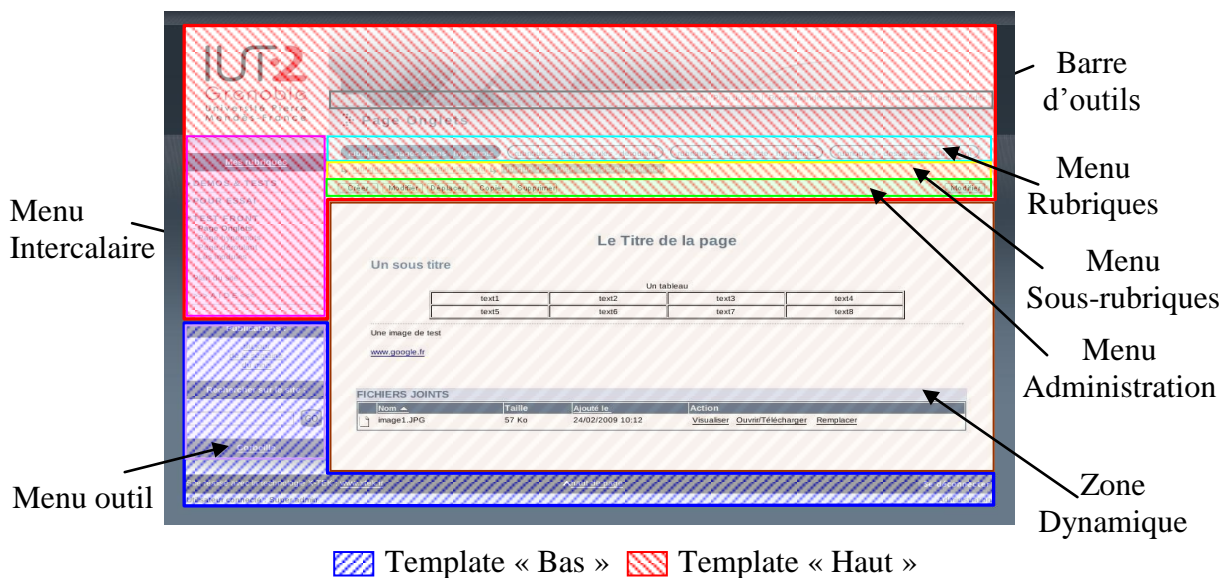


Figure 32 : Nouvelle structure de page

Avec cette nouvelle structure de page, nous pensions avoir répondu à cette problématique de respect d'un ordonnancement logique des contenus. Cependant, après avoir recherché une solution à un problème différent auprès de la liste Accessiweb, une nouvelle interprétation de ce critère d'accessibilité m'a été donnée, ce qui a remis en cause notre solution. En effet, il semblerait qu'il faille, non seulement ne pas ordonnancer logiquement les blocs d'information entre eux afin d'avoir une restitution correcte sans mise en forme, mais aussi que l'ordonnancement logique soit scrupuleusement respecté visuellement. Lors de mes échanges à ce sujet avec Monsieur Laurent DENIS, membre de la liste Accessiweb et expert en Accessibilité du Web, il m'a été donné comme argument qu'il fallait qu'une personne aveugle exploitant une page Web puisse suivre la page en parallèle avec une personne voyante. Dans notre cas, une personne décrivant la page sans connaître la fonction de chaque bloc, décrira en premier lieu le bandeau supérieur en commençant par le logo puis par la barre d'outils. Dans un deuxième temps, il décrira instinctivement la colonne de gauche de la partie centrale, c'est-à-dire le menu intercalaire puis le menu outil puis, il passera à la grande colonne de droite de la partie centrale avec le menu « rubriques », le menu « sous-rubriques », le menu d'administration et le contenu. Il terminera par le bas de page.

Après avoir cherché vainement des solutions à l'ordonnancement des menus, nous avons décidé d'abandonner les menus « rubriques » et « sous-rubriques » séparés. Nous avons alors décidé de fusionner les menus afin de garantir le respect de ce critère d'accessibilité. Les

menus « rubriques » se trouvent ainsi intercalés dans le menu intercalaire sous la catégorie ouverte et les sous-rubriques sont intercalées dans les rubriques sous la rubrique ouverte. La Figure 33 montre le résultat obtenu. Il est à noter que nous avons fait en sorte de récupérer l'apparence des rubriques et sous-rubriques du plan du site afin de garder une cohérence dans le site.

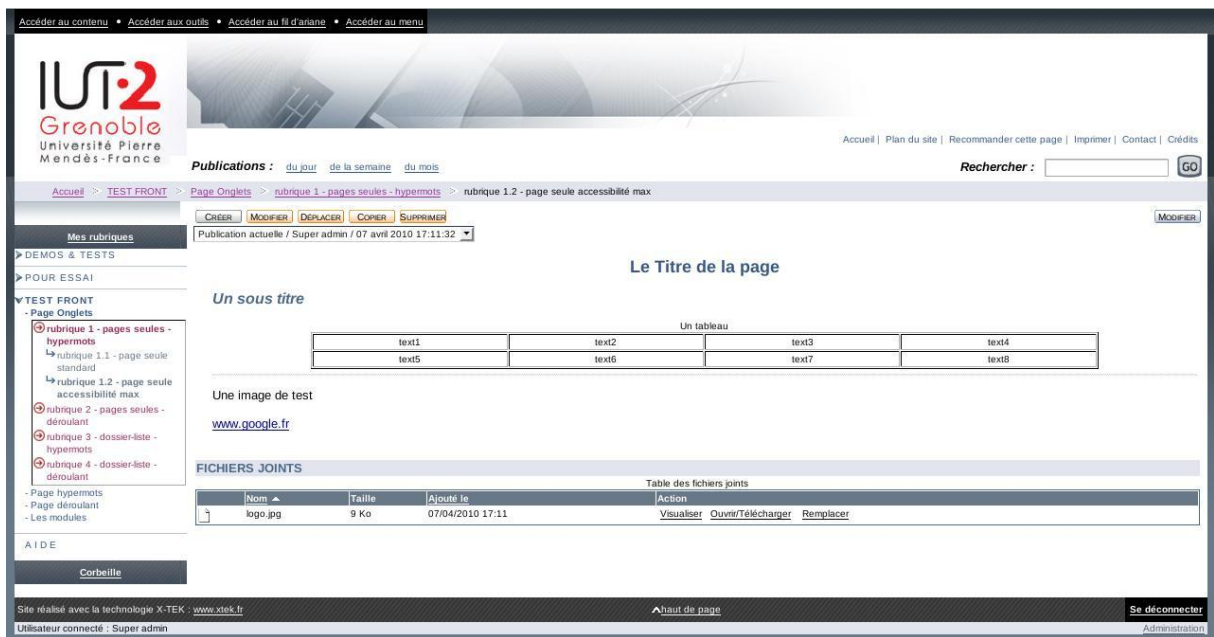


Figure 33: Structure finale de la page

### 3) Accessibilité de la présentation visuelle

Il faut que les contenus textuels soient visuellement perceptibles sans interférence dans leur lisibilité due à leur présentation. Certaines personnes ayant des déficiences cognitives, du langage, d'apprentissage ou bien encore des troubles visuels, peuvent éprouver des difficultés pour comprendre des blocs de textes sans mise en forme adaptée. Pour pallier ces problèmes, il nous est demandé :

- De ne pas justifier le texte ou fournir un mécanisme permettant de supprimer la justification à la demande.
- De limiter la largeur des blocs de texte à 80 caractères ou glyphes (40 en chinois, japonais ou coréen)

- D'assurer un interligne d'au moins 1,5 et un espacement entre paragraphes d'au moins 1,5 l'interligne du paragraphe. Sinon, il faut fournir un mécanisme permettant de passer à cette configuration à la demande.
- De faire en sorte que la barre de défilement horizontale n'apparaisse pas lorsque la fenêtre passe en plein écran.

Ce critère était trop contraignant, difficile à atteindre et vise le niveau AAA d'accessibilité. Nous nous étions donné pour consigne d'atteindre au minimum le niveau AA, c'est pourquoi nous avons concentré nos efforts sur d'autres problèmes techniques plus importants. Nous avons tout de même fait en sorte de rendre les pages de X-TEK « élastiques » afin qu'elles puissent suivre les dimensions de la fenêtre. De plus, cela nous permet aussi de mieux tirer parti des grandes dimensions des écrans actuels. Nous avons donné une taille minimale de 800 par 600 à notre page.

#### **4) Conserver l'aspect visuel des éléments de navigation et du focus**

Il faut que l'on puisse distinguer visuellement les liens du texte ainsi que le focus des éléments. Si les liens sont uniquement distinguables par la couleur, il est impératif d'avoir un contraste d'au moins 3 entre la couleur du lien et le texte environnant, ainsi qu'un autre moyen que la couleur pour distinguer le lien lorsqu'il reçoit le focus comme un soulignement, un effet de graisse ou bien encore l'ajout d'une icône.

De plus, il faut que sur tout élément pouvant recevoir le focus, l'aspect avec ou sans focus soit discernable. En standard, le focus est repéré par un encadrement en pointillé autour de l'élément, il faut donc veiller à avoir une distinction similaire.

Au niveau de X-TEK, nous avons vérifié que les liens restaient distinguables lorsqu'ils sont positionnés dans du texte. Nous sommes partis du principe que lorsqu'un lien est positionné dans un groupe de liens, sans texte adjacent, nous ne sommes pas tenus de maintenir une distinction visuelle.

Nous avons aussi vérifié que tout élément pouvant recevoir le focus soit distinguable de sa représentation sans focus.

## 5) Lisibilité en cas d'agrandissement

Il nous est demandé que le document reste lisible lorsque nous augmentons la taille du texte de 200%. Il nous faut aussi faire en sorte que les tailles des styles de caractères ne soient pas exprimées en valeur absolue mais plutôt en valeur relative, afin de permettre le dimensionnement des styles avec l'agrandissement de la taille de texte par défaut sur le navigateur.

Bien que cette partie concerne essentiellement les anciens navigateurs, nous avons exprimé tous les styles des textes en valeur relative et nous avons veillé à la lisibilité de la page lors d'un agrandissement à 200%. Ceci a été facilité par le fait d'avoir rendu les pages de X-TEK « élastiques ».

## 6) Restitution correcte des éléments masqués

Il faut que tout effet permettant de masquer du contenu, le fasse en positionnant le contenu masqué en dehors de la vue de la fenêtre et non en utilisant les instructions des feuilles de styles comme « `display :none` » ou « `visibility :hidden` ». En effet, il existerait des effets inattendus avec certains logiciels de revue d'écran ou agent utilisateur d'accessibilité lorsque l'on emploie ces instructions. Le contenu masqué deviendrait inaccessible dans certains cas. Cependant, le fait de déplacer le contenu en dehors de l'écran permettrait d'obtenir l'effet souhaité tout en restant compatible avec les logiciels en cause.

Le RGAA nous renvoie vers la technique C7 du WCAG 2.0 qui décrit la technique à mettre en œuvre. Nous l'avons alors appliquée au menu qui faisait usage de l'instruction CSS « `display :none` ».

### 4.3.7. Thématique « Structure »

Cette thématique vise à offrir aux pages une structure cohérente et une hiérarchie de titres complète. L'usage d'en-têtes de section ou de titres hiérarchisés permet à tout utilisateur, quel que soit son mode d'accès, de saisir l'organisation de la page.

Il existe 13 tests que nous pouvons regrouper en 4 sous-ensembles de cette thématique.

## 1) Hiérarchie de titre

Il faut que toutes les pages possèdent au moins un titre de hiérarchie de premier niveau que l'on obtient à l'aide de l'élément « h1 ». De plus, si le contenu de la page nécessite plus de niveaux et de sous-niveaux de hiérarchie de titre, il faut les mettre en place.

Au niveau de X-TEK, nous possédions déjà un titre de premier niveau qui ne faisait que reprendre le titre de la catégorie courante et restait donc invariant quelle que soit la rubrique ou sous-rubrique consultée. De plus, vu sa position dans le code source, le titre ne pouvait en aucun cas annoncer le contenu qui le suivait.

Nous avons alors profité de la restructuration de la page au niveau des menus, des outils et de l'apparition du fil d'Ariane, afin de repenser entièrement la structure hiérarchique des titres. Nous nous sommes alors confrontés au problème du choix des niveaux hiérarchiques pour les différentes sections de la page. Il existe un débat sur Internet dans le milieu de l'accessibilité sur l'usage du premier niveau hiérarchique dans les pages Web. Certains soutiennent que l'on ne doit utiliser qu'une seule fois l'élément « h1 » dans une page et que cet élément doit porter le titre de la page. D'autres, par contre, considèrent uniquement l'élément « h1 » comme un titre de section et que le véritable titre de la page se retrouve dans l'élément « title » en tête de page. Ces derniers se reposent sur la spécification pour étayer leur argumentaire comme le fait Monsieur Laurent DENIS dans son article « title et h1, titre de section et titre de document : le malentendu » [DENIS01] qui nous aura définitivement conforté dans notre choix d'utiliser plusieurs éléments « h1 » pour les différentes sections de notre page.

Nous avons alors créé des titres de premier niveau pour les sections suivantes :

- Section « outils » qui possède les deux sous-sections « barre des fonctions », « publications » et « rechercher ». Les publications ne sont visibles qu'en back-office.
- Section « Fil d'Ariane ».
- Section « Menus » qui possède les sous-sections « Mes rubriques », « Menu Intercalaires » et « Corbeille ». Les sous-sections « Mes rubriques » et « Corbeille » ne sont accessibles qu'en back-office.
- Section « Menu d'administration » disponible en back-office.

- Section de contenu reprenant comme intitulé le nom de la page courante. Nous rappelons que le titre de la page est constitué du nom de la plateforme concaténé au nom de la page.
- Enfin la section « Information » qui introduit le pied de page.

Nous obtenons ainsi le plan de page suivant :

```
<h1>Outils</h1>
  <h2>Barre des fonctions</h2>
  <h2>Publications :</h2>
  <h2>Rechercher :</h2>
<h1>Fil d'Ariane</h1>
<h1>Menus</h1>
  <h2>Mes rubriques</h2>
  <h2>Menu Intercalaires</h2>
  <h2>Corbeille</h2>
<h1>Menu d'administration</h1>
<h1>...titre du contenu...</h1>
<h1>Informations</h1>
```

**Figure 34 : Plan de la page**

Vu que la majorité de ces titres n'ont pas été prévus pour être intégrés dans la charte graphique de la page, nous les avons rendus invisibles en les dessinant en dehors de la zone d'affichage. Ils restent cependant parfaitement accessibles lors d'un usage sans style.

Le dernier problème que nous rencontrons provient des contributeurs. En effet, ils sont amenés à entrer du contenu à l'aide de l'éditeur de texte enrichi et peuvent, le cas échéant, déroger à la structure même de la page. Leur contenu est censé être ajouté après un élément « h1 » et doit donc utiliser uniquement des titres de hiérarchie inférieure au premier niveau.

Nous avons imaginé deux solutions à ce problème :

- Soit interdire l'usage d'élément « h1 » dans l'éditeur de texte enrichi afin de garantir qu'il ne sera jamais utilisé dans le contenu des contributeurs.
- Soit retravailler automatiquement le contenu des contributeurs afin de réadapter les titres utilisés. Toutefois, cela va à l'encontre de la notion même d'éditeur WYSIWYG car le contenu serait affiché différemment entre la phase d'édition et le rendu final à l'écran.

Nous n'avons toujours pas retenu de solution car il nous faut au préalable mettre en place un nouvel éditeur de texte enrichi.

## **2) Pertinence et cohérence des structures de la page**

Il nous faut vérifier la pertinence des titres de hiérarchie ainsi que l'absence d'interruption dans la hiérarchie. De plus, il faut vérifier le balisage correct des listes non ordonnées, des listes de définitions et des citations, ainsi que l'absence de simulation visuelle de liste non ordonnée.

Au niveau de X-TEK, nous avons vérifié la hiérarchie et l'absence de simulation visuelle de liste. Pour ce qui est des citations et définitions, le respect de la norme reposera sur les contributeurs.

## **3) Balisage des abréviations et acronymes**

Il faut baliser convenablement les abréviations et les acronymes, soit :

- En utilisant les balises adéquates telles que « abbr » pour les abréviations et « acronym » pour les acronymes.
- En utilisant un lien adjacent renvoyant vers la définition de l'abréviation ou de l'acronyme.
- En donnant directement la signification de l'abréviation ou de l'acronyme dans un texte adjacent.

- En fournissant un glossaire dans la page.

De la même manière que les citations et les définitions, le respect de la norme sera à la charge du contributeur qui devra faire usage des fonctionnalités offertes par l'éditeur de texte enrichi.

#### **4) Accessibilité des documents bureautiques**

Il faut, de même, respecter toutes ces règles de structure pour les documents bureautiques téléchargeables. Si possible, ils devront être directement accessibles sinon il faudra fournir une version accessible au format HTML.

Au niveau de X-TEK, nous ne pouvons pas contrôler le contenu des fichiers bureautiques fournis par les contributeurs. Il faudra donc qu'ils soient validés manuellement avant de les intégrer à la plateforme.

#### **4.3.8. Thématique « Tableau »**

Cette thématique vise à garantir l'accessibilité des tableaux en veillant à leur linéarisation correcte et à l'association des cellules à leur en-tête de colonne et/ou de ligne.

Il existe 10 tests que nous pouvons regrouper en 2 sous-ensembles de cette thématique.

##### **1) Information et association**

Il nous est demandé, au niveau des tableaux de données, de fournir un titre et un résumé en utilisant les éléments HTML correspondants conformément aux techniques H39 et H73. L'association d'un titre et d'un résumé au tableau permet aux utilisateurs accédant séquentiellement à la page, de saisir le sujet du tableau sans devoir le parcourir.

Il nous faut aussi baliser correctement les en-têtes des tableaux. En effet, nous rencontrons fréquemment des tableaux de données composés uniquement de cellules définies à l'aide d'éléments « td » et même pour les en-têtes de colonnes et/ou de lignes. Or, en HTML, il existe l'élément « th » qui a pour fonction de porter des informations d'en-tête. L'utilisation d'élément « th » au lieu de « td » pour les en-têtes permet aux agents utilisateurs de différencier convenablement les cellules d'en-têtes de celles de données et permet ainsi de les présenter différemment aux utilisateurs. Graphiquement, les en-têtes seront naturellement



restitués avec un effet de « graisse » et certains synthétiseurs vocaux utiliseront une autre inflexion de voix pour marquer la différence.

Enfin, il nous faut associer les cellules de données à leurs cellules d'en-têtes permettant ainsi aux utilisateurs ayant un accès séquentiel au tableau de rapidement associer la donnée qu'ils sont entrain de consulter avec l'en-tête correspondant. Pour ce faire, le RGAA nous renvoie vers les techniques H43 et H63.

La technique H63 propose de définir l'attribut « scope » des cellules « th » ou des cellules « td » utilisées en tant qu'en-tête, afin de préciser si elles sont des en-têtes de colonne ou de ligne. Les valeurs possibles pour l'attribut « scope » sont :

- « col » pour définir un en-tête de colonne
- « row » pour définir un en-tête de ligne
- « rowgroup » et « colgroup » pour définir respectivement un en-tête d'un groupe de lignes et un en-tête d'un groupe de colonnes.

La technique H43 est, quant à elle, destinée aux tableaux complexes. Elle associe les en-têtes aux cellules de données, en définissant un identifiant pour chaque en-tête et en définissant l'attribut « headers » de chaque cellule de données avec les identifiants de chaque en-tête. Nous pouvons alors avoir un, deux ou même plusieurs en-têtes pour une seule cellule.

Au niveau de X-TEK, nous avons appliqué cette thématique sur le seul tableau de données du dossier une seule page qu'est le tableau des fichiers joints. Nous avons donc modifié la ligne d'en-tête en remplaçant les cellules d'éléments « td » par des éléments « th ». Nous avons associé les cellules de données à leur en-tête de colonne en utilisant les deux techniques H43 et H63, bien que la technique H63 aurait été suffisante dans ce cas. Nous avons alors défini l'attribut « scope » en tant que colonne. Nous avons aussi défini un identifiant pour chaque en-tête et nous avons défini les attributs « headers » des cellules de données en utilisant l'identifiant de l'en-tête correspondant. Nous avons enfin rajouté un titre pour le tableau en utilisant l'élément « caption », ainsi qu'un résumé à l'aide de l'élément « summary ».

Il est à noter que nous avons effectué une première mise en conformité des tableaux, en utilisant le RGAA 1.0 dans lequel il nous était demandé de rassembler les lignes d'en-tête au

sein d'un élément « thead » et le contenu du tableau dans un élément « tbody ». Le RGAA 2.0 ne nous demande plus d'utiliser ces éléments mais nous les avons tout de même conservés car ils ne nuisent pas à l'accessibilité, bien au contraire. En effet, cette astuce permet aux agents utilisateurs de répéter les en-têtes lorsque l'on exploite un grand tableau de données avec un défilement vertical.

Nous avons aussi gardé les abréviations des en-têtes de colonne comme le demandait le RGAA 1.0. L'abréviation est obtenue en utilisant l'attribut « abbr » sur la cellule d'en-tête. Cette astuce permet d'éviter de répéter continuellement l'en-tête complet lorsque l'on exploite le tableau à l'aide du synthétiseur vocal.

## **2) Linéarisation**

Il nous est demandé de vérifier la linéarisation correcte des tableaux de mise en page, afin de conserver la cohérence entre les contenus et de vérifier l'absence de simulation de tableaux à l'aide de textes.

Au niveau de X-TEK, nous avons vérifié qu'il n'existait aucune simulation de tableaux à l'aide de textes. Nous ne pouvons pas garantir que les contributeurs n'en feront pas usage. La seule solution est de bien former les contributeurs ou bien de mettre en place un processus de validation des contenus publiés par un responsable accessibilité.

Nous avons proscrit l'utilisation de tableaux de mise en page en les remplaçant par une mise en page à l'aide de styles.

### **4.3.9. Thématique « Texte »**

Cette thématique vise à garantir l'accessibilité des contenus textuels en veillant à leur lisibilité et leur cohérence dans la page.

Il existe 10 tests que nous pouvons regrouper en 4 sous-ensembles de cette thématique.

#### **1) Equivalence des alternatives textuelles aux éléments exécutables**

Il nous est demandé de veiller à l'équivalence des alternatives textuelles aux éléments exécutables tels que les applets ou les scripts générant ou affichant un contenu important.

Nous pouvons prendre pour exemple un élément « object » permettant l’affichage d’une vidéo. Il nous faudrait, dans ce cas, décrire la vidéo tout en donnant une retranscription des dialogues conformément à la thématique multimédia.

Au niveau de X-TEK, il n’est fait usage que de scripts ne générant que peu de contenu textuel et aucun contenu visuel. Nous avons donc déjà donné des alternatives suffisantes au niveau de la thématique scripts.

## 2) Absence de syntaxes cryptiques

Il nous est demandé de veiller à l’absence de syntaxes cryptiques tels que les arts ASCII, les émoticônes et du texte en langage « leetspeak », ou bien de fournir une alternative textuelle à ces éléments. Ceci permet de garantir un accès équivalent à l’information pour l’ensemble des utilisateurs.

Les syntaxes cryptiques n’ont de sens que pour leur rendu visuel et ne sont donc pas accessibles pour les malvoyants. L’art ASCII était très utilisé au début de l’informatique lorsqu’il était impossible d’afficher autre chose que du texte. Un exemple d’art ASCII est donné en Figure 35.

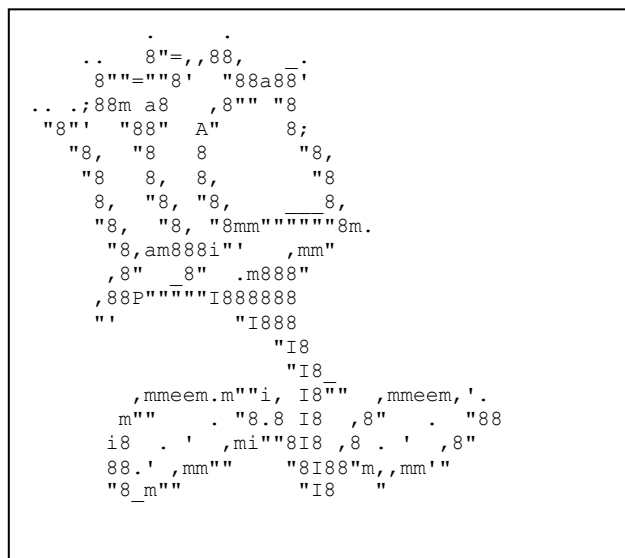


Figure 35: Fleur en Art ASCII

Tandis que le langage « leetspeak » est une manière d’écrire du texte en remplaçant les caractères par des chiffres ou bien même des symboles ASCII leurs ressemblant. Par exemple,

« leet speak » s'écrit en langage « leet speak » soit « 1337 sp34k », en codage de base, ou bien « 1337 5|\*34|< », en codage léger.

Au niveau de X-TEK, nous n'avons aucune syntaxe cryptique et nous n'avons aucun moyen de contrôler la création de contenu de ce type par les contributeurs. Il faudra, là aussi, former les contributeurs.

### **3) Accessibilité des identifications des contenus**

Il faut veiller à ne pas faire référence à du contenu par sa forme ou par sa position car ces informations sont inaccessibles aux utilisateurs malvoyants. Par exemple, il ne faut pas spécifier, dans des instructions présentes sur une page, qu'il faut cliquer sur le bouton rond pour effectuer une certaine action et sur le bouton carré pour effectuer une autre action. De même, il ne faut pas préciser que le lien à gauche permet de revenir à la page précédente ou que le lien de droite permet d'aller à la page suivante. La solution préconisée est de faire référence au contenu par du texte. Par exemple, dans le cas d'un bouton de validation ayant pour texte « Valider » permettant la validation d'un formulaire, il faut préciser dans les instructions de la page, de cliquer sur le bouton « Valider » pour déclencher la validation du formulaire.

Au niveau de X-TEK, nous avons vérifié qu'il n'était jamais fait référence à du contenu par sa forme ou sa position. Il faudra tout de même former les contributeurs sur ce point.

### **4) Lisibilité et compréhension des contenus**

Afin de rendre lisible et compréhensible les contenus textuels de nos pages, il nous est demandé d'informer l'utilisateur de tout changement de langue dans la page, de fournir une définition lorsque nécessaire comme par exemple lorsque l'on utilise du jargon, de fournir la prononciation et d'une manière générale de faire en sorte d'utiliser un style de rédaction simple et compréhensible par tous.

Au niveau de X-TEK, nous n'avons pu que vérifier qu'il n'y avait pas de changement de langue au sein de pages et qu'il n'existait pas de textes nécessitant une définition. Tous ces points seront uniquement vérifiés lors de la création de contenu par les contributeurs. Il faudra

tout de même que l'éditeur de texte enrichi permette de préciser les changements de langue et de créer des définitions.

#### **4.3.10. Thématique « Formulaire »**

Cette thématique vise à garantir l'accessibilité des formulaires en veillant à la bonne structuration des formulaires et en offrant une aide à la saisie.

Nous avons uniquement travaillé sur le formulaire du moteur de recherche car la mise en conformité de la partie saisie du module Formulaire demandait une modification en profondeur de son fonctionnement.

Il existe 16 tests que nous pouvons regrouper en 2 sous-ensembles de cette thématique.

##### **1) Structure des formulaires**

Il faut que tout élément de formulaire possède un identifiant et qu'une étiquette lui soit associée. L'association de l'étiquette sera le plus souvent effectuée à l'aide d'un élément « label » au niveau duquel l'attribut « for » aura pour valeur l'identifiant de l'élément de formulaire auquel est associé l'étiquette. Dans le cas où nous ne pouvons mettre en place un élément « label », l'étiquette prendra place dans l'attribut « title » de l'élément de formulaire et décrira la fonction exacte de l'élément. Cette astuce permet d'assurer l'association des étiquettes aux éléments de formulaire et permet ainsi aux utilisateurs ayant un accès séquentiel à la page, de retrouver l'étiquette de l'élément de formulaire en cours de consultation.

Il nous faut aussi grouper les éléments de formulaire ayant une relation entre eux et fournir une information textuelle supplémentaire au groupe. Prenons par exemple un formulaire d'inscription qui demande de fournir une multitude d'informations telles que le nom, prénom, l'adresse, le code postal, la ville et bien d'autres informations. L'adresse, le code postal et la ville sont liés ensemble et forment un ensemble que l'on pourrait nommer « adresse résidentielle ». Nous pouvons alors les regrouper dans un élément « fieldset » et fournir à l'aide de l'élément « legend » la légende « adresse résidentielle ». Ainsi l'utilisateur qui entrera dans ce groupe sera informé qu'il lui sera demandé de fournir les informations relatives à son adresse résidentielle.

De même, nous devons grouper les éléments des listes déroulantes le nécessitant comme, par exemple, pour une liste de véhicules possédant des automobiles, des camions et des motocyclettes. Nous pouvons créer trois groupes de véhicules que nous nommons : « automobiles », « camions » et « motocyclettes ». Au niveau de l'élément « select » nous aurons trois groupes formés à l'aide d'éléments « optgroup » ayant pour attribut « label » le nom du groupe correspondant. Les éléments « option » de chaque véhicule sont alors insérés dans l'élément « optgroup » correspondant à leur type de véhicule.

Au niveau de X-TEK, nous avons juste modifié le formulaire de recherche. Nous avons fourni aux éléments des identifiants et des étiquettes. Nous n'avons pas eu besoin de regrouper des éléments, ni d'intervenir sur des listes déroulantes.

## 2) Aide à la saisie et correction d'erreur

Il nous est demandé :

- Sur les champs nécessitant un contrôle de saisie, de permettre l'identification des erreurs de saisie en indiquant les champs erronés, et si nécessaires le type et/ou le format attendu.
- De préciser le caractère obligatoire de certains champs de saisie, ainsi que le type et/ou le format attendu. Ces précisions peuvent être effectuées dans des instructions avant le formulaire et répétées dans un élément « label » associé à l'élément. L'obligation peut être indiquée de manière simplifiée au niveau des champs à l'aide d'un caractère distinctif.
- De mettre en place, au minimum pour les données à caractère juridique, financier ou personnel, un mécanisme permettant de vérifier, modifier ou confirmer le formulaire avant validation.
- De mettre en place une page d'aide ou un mécanisme d'aide contextuelle à la saisie des formulaires.

Au niveau de X-TEK, le seul formulaire que nous ayons mis en conformité est le formulaire de recherche. Il ne nécessite aucune aide ou contrôle de saisie. Nous nous sommes intéressés à la partie saisie du module formulaire qui est bien plus complexe et surtout dynamique. Le

module formulaire permet déjà de spécifier le caractère obligatoire de chaque champ et peut potentiellement indiquer le type et/ou format attendu de chaque champ à l'aide du paramètre commentaire disponible à la création du masque de formulaire. Toutefois, nous ne possédons pas de mécanisme de contrôle de saisie ni de validation de formulaire. La mise en place de ce type de mécanisme nécessite d'importantes modifications du fonctionnement des formulaires. C'est pourquoi, nous avons préféré nous concentrer en priorité sur les autres problèmes d'accessibilité, afin de rendre le « dossier une seule page » accessible bien que le module « Formulaire » fasse partie intégrante du module « Page » gérant le « dossier une seule page ».

#### **4.3.11. Thématique « Cadre »**

La thématique « cadre » ne comporte que 2 tests visant à garantir l'absence de cadre non titré et que ce titre soit pertinent.

Au niveau de X-TEK, nous avons très peu d'usage de cadres. Ils sont utilisés uniquement dans le module « page Web » et lors de l'utilisation de l'éditeur de texte enrichi. Ils n'apparaissent pas dans le module « dossier une seule page ».

#### **4.3.12. Thématique « Multimédia »**

Cette thématique vise à garantir l'accessibilité des médias synchronisés et de manière générale tout contenu multimédia comme les applications embarquées. De plus, elle permet de vérifier l'absence de contenu pouvant provoquer des crises chez les utilisateurs comme des flashes lumineux ou nuire à sa navigation.

Il existe 34 tests que nous pouvons regrouper en 3 sous-ensembles de cette thématique.

##### **1) Médias synchronisés**

Cette partie reprend entièrement la règle « Média temporel » du WCAG 2.0. Bien qu' X-TEK ne soit pas prévu pour diffuser des médias synchronisés, il nous faut tout de même prévoir leur usage dans la limite de leur téléchargement à partir de X-TEK ou de la présence de liens vers ce type de contenus. Nous n'avons pour ainsi dire aucun moyen, actuellement, pour

contrôler la présence de ce type de contenu ce qui implique que le respect de cette thématique reposera essentiellement sur la formation des contributeurs.

Dans cette partie, il nous est demandé de fournir des alternatives aux contenus préenregistrés seulement audios, seulement vidéos et aux contenus vidéos contenant des flux audios. Il nous est aussi demandé de fournir des alternatives aux contenus audios et vidéos en direct.

*a) Contenus préenregistrés*

Quel que soit le handicap, il nous est demandé de fournir une alternative textuelle aux contenus préenregistrés. Cette alternative peut, selon le cas, être assez conséquente en fournissant, en plus de la retranscription des dialogues, la description des éléments visuels importants, des bruits, des musiques ou bien même des émotions, intonations et informations complémentaires. L'alternative textuelle peut être accessible selon le type de média et le contenu porté par le média, soit :

- Téléchargeable ou consultable à partir d'un lien suivant directement le lien permettant de télécharger le média.
- Embarqué dans l'élément permettant la lecture du média comme dans le corps d'un élément « object » ou bien dans l'élément « noembed » fournissant une alternative à un élément « embed ».
- En tant que sous-titre synchronisé au média, à l'aide de technologies telles que le langage SMIL

De plus, selon le handicap des utilisateurs, il faut prévoir des solutions annexes. Comme pour les utilisateurs souffrant d'un handicap visuel, il faut prévoir une audio description des médias visuels. De même pour certains utilisateurs souffrant de déficiences auditives, il faut prévoir, selon les cas, une version en langue des signes.

Au niveau de X-TEK, nous pouvons seulement fournir un lien permettant de télécharger une alternative textuelle à un média synchronisé téléchargeable à partir de la page. L'accessibilité des médias synchronisés repose donc sur les contributeurs.



**b)           *Contenus diffusés en direct***

Le RGAA demande de fournir un sous-titrage synchronisé aux médias diffusés en direct. Ceci est difficile à mettre en œuvre et le respect de ce critère d'accessibilité nous amène à un niveau d'accessibilité triple « A ». Il n'existe pas de solutions pré-établies, cependant certaines pistes nous sont fournies comme :

- La retranscription textuelle de notes sténographiques saisies par un opérateur pendant le direct et synchronisées au média.
  
- La reconnaissance vocale des dialogues en temps réel.

X-TEK n'a pas vocation à diffuser des médias en direct, nous n'avons donc pas envisagé de solution pour ce type de média.

**2)           *Accessibilité des éléments « object », « applet » et « embed »***

Il nous est demandé de faire en sorte que les éléments « object », « applet » et « embed » soient accessibles.

Pour ce faire, nous devons fournir une alternative en l'insérant dans le corps de l'élément comme il nous l'est proposé dans les techniques G58, G69, G158, G159, H27, H35 et H53. Dans l'exemple suivant en Figure 36, tiré d'un des exemples de la technique H27, nous pouvons tirer parti de la capacité des éléments « object » à inclure d'autres éléments « object ». Dans cet exemple, le premier élément « object » a pour fonction de déclencher une applet Java montrant un phénomène de pression dans un ballon. Si l'applet Java n'est pas accessible, il passera alors au deuxième élément « object » qui donne une alternative sous forme d'une vidéo. Si la vidéo n'est pas accessible, il passera à l'alternative suivante sous forme d'image. Finalement, il restera l'alternative textuelle si rien n'a fonctionné.

```
<object classid="java:Press.class" width="500" height="500">
  <object data="Pressure.mpeg" type="video/mpeg">
    <object data="Pressure.gif" type="image/gif">
      Comme la temperature augmente, les molecules à l'intérieur du ballon...
    </object>
  </object>
</object>
```

**Figure 36: Inclusion d'éléments "object"**

Il faut aussi veiller à l'accessibilité des éléments programmables avec les aides techniques. Ainsi, il faut faire en sorte d'utiliser les API d'accessibilité des langages employés comme en Java ou en Flash. De plus, il faut rendre les éléments programmables utilisables au clavier afin de garantir leur accessibilité quel que soit le périphérique employé. Dans le cas où nous ne pourrions pas fournir une application accessible, il nous faut fournir une alternative accessible à cette application.

Au niveau de X-TEK, nous n'avons pas d'usage d'éléments « object », « applet » ou « embed ». Nous avons juste vérifié que ces éléments n'étaient pas présents.

### 3) Crises et nuisances

Il faut interdire tout ce qui pourrait nuire au confort de l'exploitation de la page ou pourrait provoquer une crise chez l'utilisateur. En effet, certaines personnes, tels les épileptiques, sont sensibles aux flashes lumineux et aux changements brusques de luminosité. L'apparition de flashes à certaines fréquences peut provoquer chez eux une crise d'épilepsie. D'autres ont des problèmes de concentration ou de perception, si bien que tout mouvement sur la page nuit à leur capacité de lecture. Enfin, certaines personnes peuvent être gênées par un fond sonore présent sur la page, comme les utilisateurs faisant usage d'un logiciel de revue d'écran qui retranscrit le contenu vocalement. Un fond sonore trop fort risquerait de masquer la retranscription vocale de la page. D'une manière générale, il faut veiller, pour toute bande son, à avoir la piste des dialogues 20 décibels plus haute que le fond sonore.

Il faut donc interdire tout flash, clignotement, mouvement et fond sonore sur la page, ainsi que tout déclenchement automatique de ce type de comportement sans possibilité pour l'utilisateur de les arrêter. Il nous est alors demandé de vérifier l'absence d'éléments « blink »,

« marquée » et « bgsound », ainsi que l'absence de ce type d'effets produits par des styles ou du code javascript.

Au niveau de X-TEK, nous n'avons aucun phénomène de flash, de mouvement ou de son. Nous avons simplement vérifié qu'il n'était pas fait usage de ce genre d'effets.

## **4.4. Recherche d'un nouvel éditeur de texte enrichi**

Après avoir déroulé plusieurs fois les différents tests des 12 thématiques du RGAA 2.2, nous sommes parvenus à obtenir un module « dossier une seule page » valide. Cependant, un composant primordial de ce module pose toujours problème. L'éditeur de texte enrichi ne génère toujours pas du code accessible. Nous nous sommes alors mis en quête d'un nouvel éditeur.

Il existe une multitude d'éditeurs de texte enrichi mais très peu ont été développés en respectant les règles d'accessibilité. Nous sommes tout de même parvenus à trouver 3 éditeurs de texte enrichi respectant les règles d'accessibilité. Nous avons alors retenu :

- TinyMCE : Développé par Moxiecode Systems, il s'agit d'un éditeur de texte enrichi reposant sur le couple javascript et HTML. Il est donc indépendant de la plateforme utilisée. TinyMCE est très utilisé dans le milieu des CMS Open Source qui l'intègrent généralement comme éditeur WYSIWYG de base. Il est distribué sous licence GPL 2.1.
- CKEditor : Il s'agit encore d'un éditeur de texte enrichi reposant sur le couple javascript et HTML. Il est développé par CKSource ; il est open source. CKEditor est en fait la version 3 de FCKEditor déjà utilisée par X-TEK.
- XStandard : Il s'agit d'un éditeur de texte enrichi propriétaire développé par Belius Technology disponible en deux versions. L'une dite professionnelle possédant toutes les fonctionnalités et l'autre dite allégée mais gratuite. Bien que fonctionnant avec les principaux navigateurs actuels, il reste dépendant de la plateforme utilisée. Nous ne pouvons l'installer que sous Windows et Max OS X à partir de la version 10.3.9 car l'éditeur est en fait un programme s'exécutant sur le système hôte et intégré au sein

des navigateurs. Il offre d'excellentes fonctionnalités d'accessibilité et d'intégration avec les aides techniques.

Il nous reste à faire notre choix entre ces différents éditeurs WYSIWYG.

Nous pouvons d'office écarter XStandard pour deux raisons. XStandard est dépendant de la plateforme utilisée et nécessite, dans tous les cas, une installation préalable sur le poste client. Dans un environnement maîtrisé il aurait été la meilleure solution. L'autre raison est qu'il s'agit d'une application payante et bien qu'une version allégée gratuite existe, elle ne permet pas de répondre à l'ensemble des besoins de personnalisation de l'éditeur nécessaire pour X-TEK comme la possibilité de restreindre les fonctionnalités offertes ou bien la présence de la fonction de glissé déposé vers l'éditeur de texte enrichi.

En ce qui concerne les deux autres éditeurs de texte enrichi, notre préférence allait pour CKEditor qui, lors de notre évaluation, offrait plus de fonctionnalités d'accessibilité que TinyMCE en terme d'alternative textuelle. Il possédait malheureusement une lacune au niveau des tableaux de données en ne permettant pas de définir de cellules d'en-têtes.

Au moment de la rédaction de ce mémoire, les problèmes de tableaux ont pratiquement disparu. Nous pouvons maintenant définir des lignes et/ou des colonnes d'en-tête et l'attribut « scope » du tableau est automatiquement positionné sur la valeur adéquate. D'un autre côté, TinyMCE a largement rattrapé son retard, ce qui nous laisse libre d'utiliser l'éditeur de texte enrichi qui nous convient le mieux entre TinyMCE et CKEditor.

## **4.5. Présentation des résultats**

Au terme de cette mise en conformité du module « dossier une seule page », nous avons effectué une réunion avec les principaux acteurs et décideurs du projet X-TEK que sont messieurs RAMEAU Gérard, VASSOILLE Marc et GIRAUDIN Eric.

Nous avons exposé l'ensemble des modifications apportées et avons défendu notre solution de regroupement des menus « rubriques » et « sous-rubriques » au sein du menu principal afin de respecter les critères d'ordonnancement logique des informations. La modification a finalement été acceptée en partie grâce au fait que le menu principal est resté assez compact. En effet, la plus grande crainte des membres était de voir le menu vertical grandir

démesurément, ce qui aurait eu pour effet de nuire au confort de navigation. La compacité du menu a été conservée en reproduisant la fonctionnalité de menu repliable existant sur les catégories, en l'appliquant sur les menus « rubriques » et « sous-rubriques ».

Les participants ont apprécié les nouvelles fonctionnalités telles que les informations contextuelles apparaissant lors du survol d'un élément actif, et la mise en forme élastique de la page, ajoutant un plus à l'ergonomie de X-TEK.

## **4.6. Synthèse**

Dans ce chapitre, nous avons présenté le fonctionnement de X-TEK au niveau de son architecture, de sa partie graphique et de son code source. Nous avons constaté que le module logiciel constituant le module « dossier une seule page » englobe en fait les modules « dossier menu », « dossier liste » et « formulaire en ligne ».

Après avoir introduit le déroulement de la mise en conformité, nous avons présenté les différentes thématiques de test du RGAA 2.2 que nous avons appliquées lors de cette mise en conformité, ainsi que les solutions que nous avons apportées.

Nous avons terminé par notre recherche d'un nouvel éditeur de texte enrichi et la présentation de nos résultats à l'équipe X-TEK.

## 5. Conclusion

### 5.1. Rappel des objectifs

Notre principal objectif était de rendre X-TEK accessible par rapport aux recommandations du RGAA 2.2. Le périmètre de la mise en conformité s'est ensuite restreint au module « dossier une seule page ». Une autre demande avait été faite afin de conserver l'apparence générale de l'application et un fonctionnement identique.

### 5.2. Travail réalisé

Dans un premier temps, nous avons réalisé un audit d'accessibilité de l'outil X-TEK selon le WCAG 1.0 et le RGAA 1.0, qui nous a permis de mettre en valeur la modularité de l'outil et de faire ressortir une grande partie des erreurs présentes. Nous avons pu isoler les erreurs génériques communes à l'ensemble de l'application, ainsi que celles appartenant à chacun des modules testés. Nous en avons tiré des recommandations que nous avons soumises aux membres du projet X-TEK. Au terme de la réunion de présentation de nos résultats, nous avons défini un plan de mise en conformité de l'application qui se limitera, dans un premier temps, au module « dossier une seule page ».

Dans un deuxième temps, nous sommes passés à la mise en conformité du dossier une seule page selon le RGAA 1.0 puis, après parution du nouveau référentiel, selon le RGAA 2.2. Nous avons déroulé l'ensemble des tests manuellement en nous aidant ponctuellement d'outils d'analyse des couleurs. Bien que nous nous soyons confrontés à un problème de présentation des menus nous obligeant, à posteriori, à abandonner l'objectif de respect de l'apparence de l'application et de son fonctionnement, nous sommes arrivés à résoudre l'ensemble des erreurs d'accessibilité inhérentes au module « dossier une seule page ». Nous avons aussi sélectionné un nouvel éditeur de texte enrichi qui devrait remplacer l'actuel éditeur non conforme. Pour finir, nous avons présenté la nouvelle plateforme avec sa nouvelle charte graphique et son nouveau dossier une seule page conforme.

Concrètement, nous avons rendu accessible les modules « dossier une seule page », « dossier menu » et « dossier liste ». De plus, nous avons mis en conformité le plan du site ainsi que la

partie consultation du module « formulaire ». Bien qu'elles ne soient plus utiles dans la nouvelle plateforme accessibilité, il est à noter que nous avons rendues accessibles les différentes présentations des rubriques et sous-rubriques.

Le travail restant à réaliser portera sur les modules « page Web externe », « calendrier », « forum », « messagerie interne », « trombinoscope », « liste des utilisateurs » et « flux rss ». Il faudra aussi achever la mise en conformité du module « formulaire » pour sa partie saisie.

En l'état actuel, notre plateforme peut servir de base au développement de nouvelles plateformes ou à la migration de plateformes existantes. Pour ce faire, il suffira d'adapter la charte graphique que nous avons définie, aux nouveaux besoins. Cependant, il faudra prendre garde, lors de l'adaptation, de se conformer au WCAG 2.0 car toute modification risque de générer des soucis d'accessibilité.

### **5.3. Perspectives**

La mise en conformité de X-TEK est loin d'être terminée et les modifications apportées ne permettent plus de garantir une migration aisée des plateformes en cours de production vers des plateformes accessibles. Il est à noter que les chartes graphiques de toutes les plateformes devront être modifiées et nécessiteront une analyse d'accessibilité. Il faudrait donc, avant de poursuivre la mise en conformité, tenter la migration d'une plateforme assez proche de la version de base utilisée, pour créer la plateforme accessible, afin de définir un processus de migration.

Dans un deuxième temps, il serait judicieux de mettre en production le nouvel éditeur de texte enrichi et de terminer la mise en conformité des formulaires, avant de passer à la mise en conformité des autres modules.

En ce qui concerne les modules applicatifs tels que la messagerie, le forum ou le calendrier, il serait peut-être plus intéressant d'envisager, selon la complexité des modules, de se mettre en quête de nouveaux modules respectant l'accessibilité plutôt que d'essayer de les rendre conformes. Il est possible que l'intégration d'un nouveau module soit moins contraignante que la modification d'un module existant.

Enfin, une recommandation importante et valable pour tout projet de mise en conformité d'accessibilité : le webmestre de chaque plateforme X-TEK devra mettre l'accent sur la formation de ses rédacteurs de contenu et devra s'assurer au fil du temps, qu'en pratique, les directives sont respectées.

### **5.4. Bilan personnel**

Ce stage fut pour moi une expérience enrichissante qui m'a permis de me confronter avec le milieu universitaire et de mettre en valeur mes facultés d'adaptation, de travail en équipe, et d'autonomie.

Professionnellement, j'ai beaucoup appris au contact de Monsieur Marc VASSOILLE concernant les CMS et plus généralement les technologies Web. J'ai acquis de nouvelles compétences en langage CSS et javascript, ainsi que remis à jour mes connaissances vieillissantes en langage PHP et en base de données.

Parallèlement, j'ai pu découvrir le fonctionnement d'un service informatique conséquent où règne une ambiance chaleureuse et amicale. Le contact avec l'équipe de ce service m'a profondément stimulé.

Je peux terminer en précisant que j'ai apprécié ce stage et je suis sûr que tout ce que j'ai appris ces douze derniers mois me servira, à coup sûr, dans ma vie professionnelle. Je tiens à remercier, une dernière fois, le service informatique de l'IUT2 pour son accueil et tout particulièrement Monsieur Marc VASSOILLE pour son soutien et son amitié.





## 6. Bibliographie

- [ALSAC1] Alsacreations, « Alsacreations », disponible sur : <http://www.alsacreations.com/>
- [RGAA10] DGME, « Référentiel Général d'Accessibilité pour les Administrations 1.0 », 24 novembre 2008, disponible sur :  
[http://www.references.modernisation.gouv.fr/sites/default/files/RGAA\\_version\\_1\\_0.pdf](http://www.references.modernisation.gouv.fr/sites/default/files/RGAA_version_1_0.pdf)
- [RGAA22] DGME, « Référentiel Général d'Accessibilité pour les Administrations 2.2 », 30 septembre 2009, téléchargeable à partir de :  
<http://references.modernisation.gouv.fr/rgaa-accessibilite>
- [DENIS01] Laurent DENIS, « titre et h1, titre de section et titre de document: le malentendu », disponible sur :  
<http://www.blog-and-blues.org/weblog/2004/11/03/331-titre-et-h1-titre-de-section-et-titre-de-document-le-malentendu>
- [OPENW1] OpenWeb, « OpenWeb.eu.org », disponible sur : <http://openweb.eu.org/>
- [WCAG10] :W3C, « Web Content Accessibility Guidelines 1.0 », 5 mai 1999, disponible sur :  
<http://www.w3.org/TR/WCAG10/>
- [WCAG20] :W3C, « Web Content Accessibility Guidelines 2.0 », 11 décembre 2008, disponible sur : <http://www.w3.org/TR/WCAG20/>
- [UWCAG2] W3C, « A guide to understanding and implementing Web Content Accessibility Guidelines 2.0 », 11 décembre 2008, disponible sur :  
<http://www.w3.org/TR/UNDERSTANDING-WCAG20/>
- [TWCAG2] W3C, « Techniques and Failures for Web Content Accessibility Guidelines 2.0 », 11 décembre 2008, disponible sur : <http://www.w3.org/TR/WCAG20-TECHS/>



## 7. Annexes

### 7.1. Annexe 1 : Points de contrôle du WCAG 1.0

1. Fournir des alternatives équivalentes au contenu visuel et auditif		
Points de contrôle		Priorité
1.1	Fournir un équivalent textuel à chaque élément non-textuel	1
1.2	Fournir des liens textes redondants pour chaque région active d'une carte cliquable côté serveur	1
1.3	Jusqu'à ce que les agents utilisateurs soient en mesure de lire automatiquement à haute voix l'équivalent textuel d'une piste visuelle, fournir une description auditive des informations importantes de la piste visuelle d'une présentation multimédia	1
1.4	Pour toute présentation multimédia temporisée, synchroniser les alternatives équivalentes avec la présentation	1
1.5	Jusqu'à ce que les agents utilisateurs soient en mesure de restituer les équivalents textuels des liens des cartes cliquables côté client, fournir des liens textuels redondants pour chaque région active d'une carte cliquable côté client	3

2. Ne pas s'en remettre exclusivement aux couleurs		
Points de contrôle		Priorité
2.1	S'assurer que toute information convoyée par des couleurs est également accessible sans couleur	1
2.2	S'assurer que la combinaison de couleurs entre le premier plan et l'arrière-plan utilise suffisamment de contraste lorsqu'elle est utilisée par des personnes souffrant d'un déficit de perception des couleurs ou sur un écran noir et blanc	2

3. Utiliser le balisage et les feuilles de style et cela, de façon appropriée		
Points de contrôle		Priorité
3.1	Quand un langage de balisage approprié existe, utiliser des balises plutôt que des images pour convoier l'information	2
3.2	Créer des documents qui sont validés par des grammaires formelles publiées	2
3.3	Utiliser des feuilles de style pour contrôler la mise en page et la présentation	2
3.4	Utiliser des unités relatives plutôt qu'absolues dans les valeurs d'attributs du langage et les valeurs de propriétés des feuilles de style	2
3.5	Utiliser les éléments d'en-tête pour convoier la structure du document, et les utiliser en conformité avec les spécifications	2
3.6	Marquer les listes et les éléments de listes correctement	2
3.7	Baliser les citations. Ne pas utiliser le balisage correspondant aux citations pour obtenir des effets de présentation comme l'indentation	2

4. Clarifier l'utilisation du langage naturel		
Points de contrôle		Priorité
4.1	Identifier clairement les changements survenus dans le langage naturel du texte d'un document et dans les équivalences textuelles	1
4.2	Spécifier la forme complète de toutes les abréviations ou acronymes employés dans le document lors de la première utilisation	3
4.3	Identifier le langage naturel principal du document	3

5. Créer des tableaux qui se transforment de façon élégante		
Points de contrôle		Priorité
5.1	Pour les tableaux de données, identifier les en-têtes de lignes et de colonnes	1
5.2	Pour les tableaux de données qui ont deux niveaux logiques d'en-tête de colonne ou de ligne ou plus, utiliser des balises pour associer les cellules de données et les cellules d'en-tête	1
5.3	Ne pas utiliser les tables pour la mise en page, à moins qu'elles n'aient un sens lorsqu'elles sont déchiffrées en mode linéaire. Autrement, si la table n'a pas de signification, prévoir une version alternative	2
5.4	Dans le cas où une table est employée pour la mise en page, il ne faut pas utiliser de balises structurelles dans un but de formatage visuel	2
5.5	Créer des sommaires pour les tableaux	3
5.6	Prévoir des abréviations pour les étiquettes d'en-têtes	3

6. S'assurer que les pages qui contiennent de nouvelles technologies se transforment de façon élégante		
Points de contrôle		Priorité
6.1	Organiser les documents de manière à ce qu'ils puissent être lus sans feuilles de style. Par exemple, lorsqu'un document HTML est restitué sans la feuille de style lui étant associée, il doit rester lisible	1
6.2	S'assurer que les équivalences pour le contenu dynamique soient mises à jour lorsque ledit contenu dynamique change	1
6.3	S'assurer que les pages soient visibles lorsque les scripts, les applets ou autres artefacts programmables sont désactivés ou non supportés. Lorsque ce n'est pas possible, fournir une information équivalente sur une page alternative	1
6.4	Pour les scripts et les applets, faire en sorte que les gestionnaires d'événements soient indépendants du dispositif d'entrée	2
6.5	S'assurer que le contenu dynamique reste accessible, ou fournir une présentation alternative de la page	2

7. Assurer à l'utilisateur le contrôle des changements du contenu lorsque ce dernier varie dans le temps		
Points de contrôle		Priorité
7.1	Jusqu'à ce que les agents utilisateurs permettent à l'utilisateur de contrôler les changements brusques de luminosité, il convient d'éviter de causer de tels changements sur l'écran	1
7.2	Jusqu'à ce que les agents utilisateurs permettent de désactiver le clignotement, éviter de faire clignoter le contenu	2
7.3	Jusqu'à ce que les agents utilisateurs permettent de geler le contenu mobile, éviter les mouvements sur les pages	2
7.4	Jusqu'à ce que les agents utilisateurs permettent de stopper les mises à jour, éviter de créer des pages se mettant à jour automatiquement	2
7.5	Jusqu'à ce que les agents utilisateurs permettent de désactiver la fonction de redirection automatique, ne pas utiliser de commandes pour rediriger automatiquement les pages. Configurer plutôt le serveur pour accomplir cette fonction	2

8. Assurer un accès direct aux interfaces utilisateur intégrées		
Points de contrôle		Priorité
8.1	Concevoir les éléments programmables tels que scripts et appliquestes de manière à ce qu'ils puissent être directement accessibles ou compatibles avec les différentes technologies d'assistance aux utilisateurs	1

9. Concevoir en respectant l'indépendance par rapport au périphérique		
Points de contrôle		Priorité
9.1	Prévoir des cartes cliquables côté client au lieu de cartes côté serveur, sauf lorsque les régions ne peuvent être définies par des formes géométriques disponibles	1
9.2	S'assurer que tout élément qui possède sa propre interface peut être activé d'une manière indépendante du dispositif	2
9.3	En ce qui concerne les scripts, il importe de spécifier les gestionnaires d'événements logiques plutôt que des gestionnaires d'événements dépendants de l'interface	2
9.4	Développer un ordre logique de tabulation, pour les liens, éléments de formulaires et objets	3
9.5	Prévoir des raccourcis clavier pour les liens importants	3

10. Utiliser des solutions intermédiaires		
Points de contrôle		Priorité
10.1	Jusqu'à ce que les agents utilisateurs permettent aux utilisateurs de fermer les fenêtres générées, ne pas produire de fenêtre successive ou à ouverture automatique, et ne pas modifier la fenêtre active sans prévenir l'utilisateur	2
10.2	Jusqu'à ce que les agents utilisateurs supportent les associations explicites entre étiquettes et contrôles de formulaires, s'assurer que les étiquettes sont correctement positionnées pour tous les contrôles de formulaire avec étiquettes implicitement associées	2
10.3	Jusqu'à ce que les agents utilisateurs puissent restituer des textes adjacents correctement, prévoir une alternative en mode texte linéaire pour toutes les tables qui formatent le texte en colonnes parallèles et qui ajustent le texte sur la largeur de la colonne	3
10.4	Jusqu'à ce que les agents utilisateurs puissent gérer correctement les contrôles vides, placer du texte pour occuper l'espace dans les champs vides des formulaires	3
10.5	Jusqu'à ce que les agents utilisateurs puissent gérer correctement les liens hypertextes adjacents, insérer entre ces liens des caractères imprimables non hypertextes	3

11. Utiliser des technologies et directives W3C		
Points de contrôle		Priorité
11.1	Utiliser les technologies du W3C lorsqu'elles sont disponibles et adaptées à une tâche. Utiliser les dernières versions supportées	2
11.2	Eviter d'utiliser les options des technologies du W3C qui ne sont plus supportées	2
11.3	Fournir des informations de manière à ce que les utilisateurs puissent recevoir les documents selon les préférences qu'ils ont spécifiées	3
11.4	Si, malgré vos efforts vous ne parvenez pas à produire une page accessible, créez un lien vers une autre page accessible, qui utilise les technologies du W3C, qui présente une information (ou fonctionnalité) équivalente et qui est mise à jour aussi régulièrement que la page (originale) inaccessible	1

12. Fournir des informations de contexte et d'orientation		
Points de contrôle		Priorité
12.1	Donner un titre à chaque cadre pour faciliter l'identification et la navigation entre cadres	1
12.2	Décrire l'objectif des cadres et la manière dont les cadres interagissent les uns avec les autres, si ce n'est pas évident à la seule lecture des titres	2
12.3	Lorsque c'est approprié, diviser les grands blocs d'information en groupes plus petits et plus facilement manipulables	2
12.4	Associer les étiquettes avec leurs éléments de contrôle de manière explicite	2

13. Fournir des mécanismes de navigation clairs		
Points de contrôle		Priorité
13.1	Identifier clairement la cible de chaque lien	2
13.2	Prévoir des meta-données pour ajouter des informations d'ordre sémantique aux pages et aux sites	2
13.3	Fournir des informations concernant la mise en page générale d'un site	2
13.4	Utiliser les mécanismes de navigation de manière cohérente	2
13.5	Prévoir des barres de navigation pour mettre en avant et donner accès aux mécanismes de navigation	3
13.6	Regrouper les liens de même nature, identifier le groupe (pour les agents utilisateurs) et, jusqu'à ce que les agents utilisateurs le permettent, donner un moyen de s'affranchir du groupe	3
13.7	Si l'on prévoit des fonctions de recherche, il convient de rendre possibles plusieurs types de recherches, correspondant à différents niveaux de compétences et à différents choix	3
13.8	Placer des informations distinctives au début des en-têtes, des paragraphes, des listes, etc.	3
13.9	Fournir des informations sur les regroupements de documents	3
13.10	Prévoir des moyens pour s'affranchir de l'art ASCII prenant plusieurs lignes	3

14. S'assurer que les documents soient clairs et simples		
Points de contrôle		Priorité
14.1	Utiliser le langage le plus clair et le plus simple possible, adapté au contenu de votre site	1
14.2	Associer du contenu visuel ou audio au texte, lorsque cela peut faciliter la compréhension d'une page	3
14.3	Créer un style de présentation cohérent pour toutes les pages	3





## 7.2. Annexe 2 : Critères de succès du WCAG 2.0

Principe 1 : perceptible		
Directives		
1.1 Les équivalents textuels		
Critères de succès		Niveau
1.1.1	Contenu non textuel	A
1.2 Média temporel		
Critères de succès		Niveau
1.2.1	Contenu seulement audio ou vidéo (pré-enregistré)	A
1.2.2	Sous-titres (pré-enregistrés)	A
1.2.3	Audio-description ou version de remplacement pour un média temporel (pré-enregistré)	A
1.2.4	Sous-titres (en direct)	AA
1.2.5	Audio-description (pré-enregistrée)	AA
1.2.6	Langue des signes (pré-enregistrée)	AAA
1.2.7	Audio-description étendue (pré-enregistrée)	AAA
1.2.8	Version de remplacement pour un média temporel (pré-enregistrée)	AAA
1.2.9	Seulement audio (en direct)	AAA
1.3 Adaptable		
Critères de succès		Niveau
1.3.1	Information et relations	A
1.3.2	Ordre séquentiel logique	A
1.3.3	Caractéristiques sensorielles	A
1.4 Distinguable		
Critères de succès		Niveau
1.4.1	Utilisation de la couleur	A
1.4.2	Contrôle du son	A
1.4.3	Contraste (minimum)	AA
1.4.4	Redimensionnement du texte	AA
1.4.5	Texte sous forme d'image	AA
1.4.6	Contraste (amélioré)	AAA
1.4.7	Arrière-plan sonore de faible volume ou absent	AAA
1.4.8	Présentation visuelle	AAA
1.4.9	Texte sous forme d'image (sans exception)	AAA

Principe 2 : utilisable		
Directives		
2.1 Accessibilité au clavier		
	Critères de succès	Niveau
2.1.1	Clavier	A
2.1.2	Pas de piège au clavier	A
2.1.3	Clavier (pas d'exception)	AAA
2.2 Délai suffisant		
	Critères de succès	Niveau
2.2.1	Réglage du délai	A
2.2.2	Mettre en pause, arrêter, masquer	A
2.2.3	Pas de délai d'exécution	AAA
2.2.4	Interruptions	AAA
2.2.5	Nouvelle authentification	AAA
2.3 Crises		
	Critères de succès	Niveau
2.3.1	Pas plus de trois flashes ou sous le seuil critique	A
2.3.2	Trois flashes	AAA
2.4 Navigable		
	Critères de succès	Niveau
2.4.1	Contourner des blocs	A
2.4.2	Titre de page	A
2.4.3	Parcours du focus	A
2.4.4	Fonction du lien (selon le contexte)	A
2.4.5	Accès multiples	AA
2.4.6	En-têtes et étiquettes	AA
2.4.7	Visibilité du focus	AA
2.4.8	Localisation	AAA
2.4.9	Fonction du lien (lien uniquement)	AAA
2.4.10	En-têtes de section	AAA

Principe 3 : compréhensible		
Directives		
3.1 Lisible		
	Critères de succès	Niveau
3.1.1	Langue de la page	A
3.1.2	Langue d'un passage	AA
3.1.3	Mots rares	AAA
3.1.4	Abréviations	AAA
3.1.5	Niveau de lecture	AAA
3.1.6	Prononciation	AAA
3.2 Prévisible		
	Critères de succès	Niveau
3.2.1	Au focus	A
3.2.2	A la saisie	A
3.2.3	Navigation cohérente	AA
3.2.4	Identification cohérente	AA
3.2.5	Changement à la demande	AAA
3.3 Assistance à la saisie		
	Critères de succès	Niveau
3.3.1	Identification des erreurs	A
3.3.2	Étiquettes ou instructions	A
3.3.3	Suggestion après une erreur	AA
3.3.4	Prévention des erreurs (juridiques, financières, de données)	AA
3.3.5	Aide	AAA
3.3.6	Prévention des erreurs (toutes)	AAA
Principe 4 : robuste		
Directives		
4.1 Compatible		
	Critères de succès	Niveau
4.1.1	Analyse syntaxique	A
4.1.2	Nom, rôle et valeur	A



## 7.3. Annexe 3 : Outils testés

Nous avons regroupé, dans cette annexe, les principaux outils testés et utilisés. Nous présentons dans une première partie les outils que nous avons retenus et utilisés. Dans une deuxième partie, nous présentons les outils que nous avons testés en application et non utilisés.

### 7.3.1. Outils testés et retenus

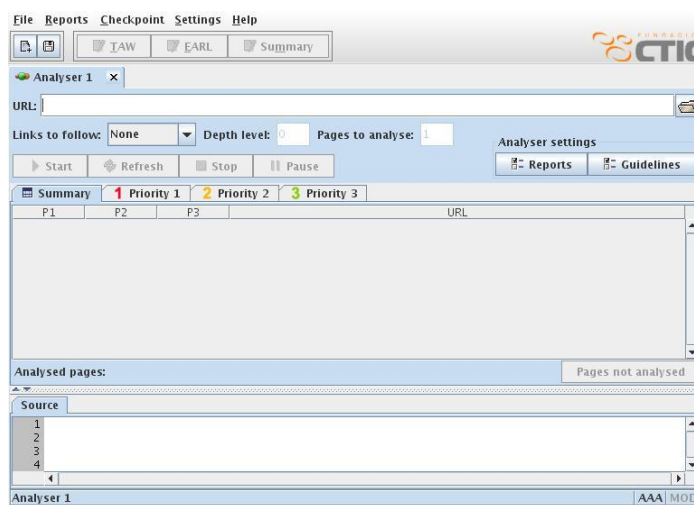
#### 1) TAW3 (Version 3.08)

TAW est un outil permettant d'évaluer l'accessibilité d'un site Web vis-à-vis des recommandations du WCAG 1.0. Il a été développé par le CTIC (centre de développement de l'information et des technologies des communications de la principauté des Asturies en Espagne).

Il offre de nombreuses possibilités de paramétrage comme :

- La personnalisation des critères d'évaluation en jouant sur le niveau d'accessibilité à tester (A, AA ou AAA) et en sélectionnant les points de contrôle à évaluer.
- L'évaluation d'un site entier ou d'une partie en jouant sur la profondeur de recherche à partir de l'URL spécifiée.

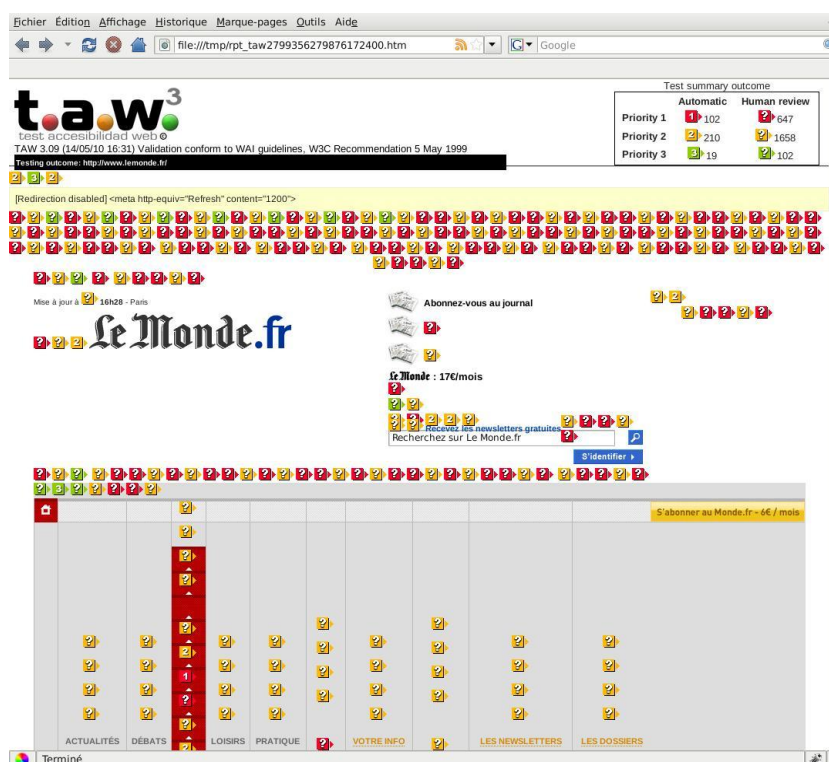
La figure suivante montre l'apparence de l'outil TAW.



L'application permet :

- L'exportation des résultats sous forme d'un rapport au format HTML avec le positionnement visuel sur la page des erreurs détectées.
- L'exportation des résultats sous forme d'un rapport au format EARL<sup>30</sup>.

La figure suivante montre un exemple de rapport HTML généré par l'outil TAW.

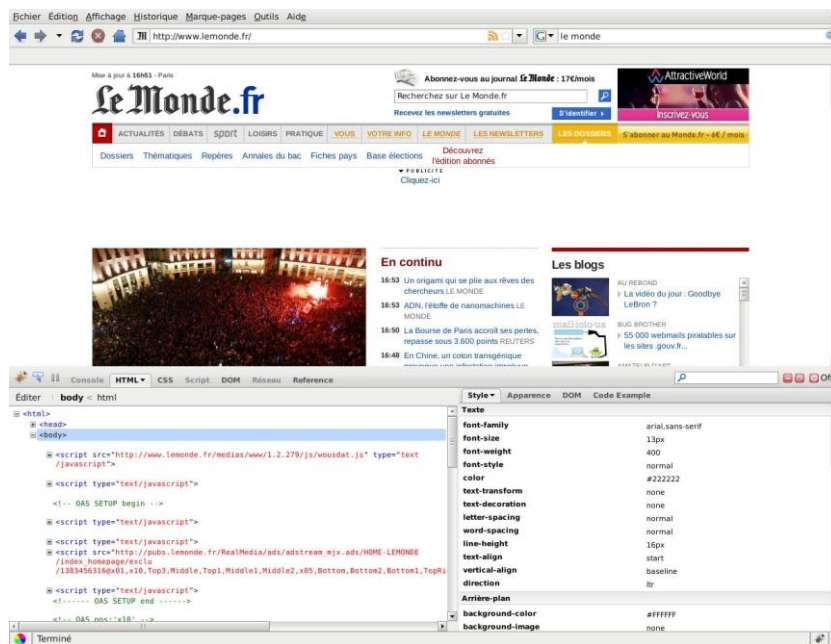


## 2) Firebug (Version 1.4.5)

Firebug est une extension pour Firefox permettant de déboguer, éditer et modifier du code HTML, CSS et JavaScript d'une page Web. Nous l'avons particulièrement utilisé lors des modifications de la charte graphique.

<sup>30</sup> EARL : Evaluation And Report Language est un langage, développé par le W3C, permettant d'exprimer les résultats de tests. Il ne se limite pas aux tests et peut s'appliquer aussi bien à des processus d'assurance qualité ou de validation

La figure suivante montre l'interface de cette extension qui apparaît en partie basse.



Firebug offre des fonctionnalités utiles comme :

- L'arborescence du code source de la page.
- L'encadrement visuel sur la page de la région correspondant à la portion de code sélectionné.
- La séquence d'héritage des styles CSS de l'élément sélectionné
- La possibilité d'éditer le code source de la page et visualiser le résultat immédiatement

### 3) Web Developer (Version 1.1.8)

Web Developer est une extension Firefox offrant de nombreuses fonctionnalités utiles pour le développement Web comme :

- La gestion des cookies, en permettant l'ajout, la suppression et la visualisation des cookies.
- La gestion de feuilles de style, en permettant l'activation et la désactivation de feuilles de style, le changement du type de média CSS et l'édition des CSS.



- La manipulation des formulaires.
- La gestion des images, en permettant, entre autre, de faire apparaître visuellement sur la page les informations des attributs « alt » et « title » des images, et d'entourer les images selon différents critères comme l'absence d'attribut « alt » ou « title ».
- L'affichage d'informations, en particulier la profondeur des tableaux et le plan du document.
- L'encadrement des éléments HTML, en particulier les balises de titre, les liens et les tableaux.
- Le redimensionnement de la page.
- L'utilisation d'outils comme la validation des feuilles de style et du code HTML.

Web Developer ne possède pas d'interface graphique ; il est intégré comme un menu de Firefox, accessible à partir d'une barre de menu ou à partir du menu apparaissant lors d'un clic du bouton droit.

Nous l'avons retenu principalement pour sa capacité à déclencher une validation du code HTML local sur le service de validation du W3C (W3C Markup Validation Service) à partir d'un raccourci clavier.

#### **4) Colour Contrast Analyser (Version 1.3 – extension Firefox)**

Colour Cotrast Analyser est une application permettant de tester les couleurs de fond et de premier plan du texte d'une page Web selon les algorithmes de l'AERT et du WCAG. Le résultat est retourné dans un rapport au format HTML. La figure suivante montre un exemple de rapport obtenu.

**Outil d'analyse des contrastes de couleurs**

Summary of Failures

	Failures
Luminosity Contrast Ratio	0
Difference in Brightness	0
Difference in Colour	3

Résultats du contraste des couleurs (pour tous les tests)

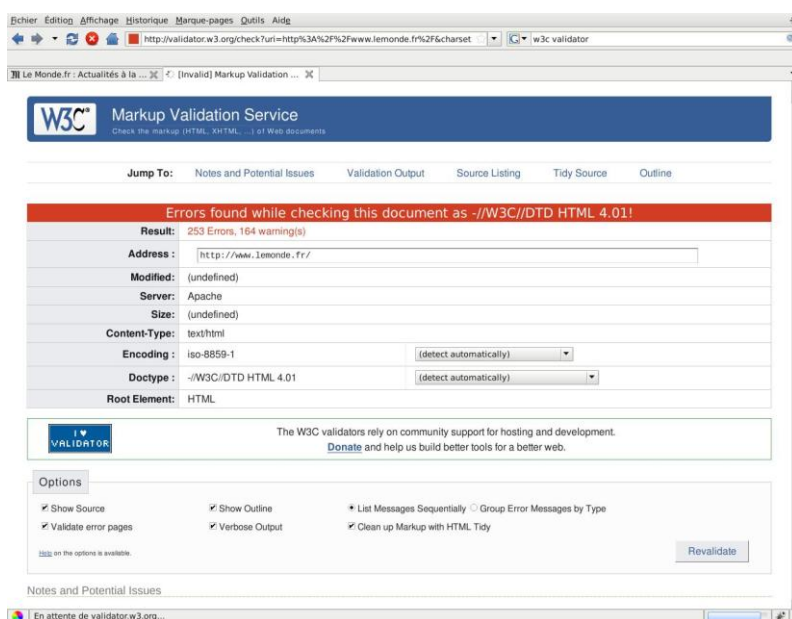
Élément	Noeuds parents	Echantillon	Couleur	Arrière-plan	Rapport de contraste de luminosité	Différence d'intensité	Différence de couleurs
A	<ul style="list-style-type: none"> <li>HTML</li> <li>BODY</li> <li>Div#container</li> <li>Div#header</li> <li>UL#skipmenu</li> <li>LI</li> </ul>	Echantillon	#000000	#ffff	21:1 (succès au niveau AAA)	255 (succès)	765 (succès)
H2	<ul style="list-style-type: none"> <li>HTML</li> <li>BODY</li> <li>Div#container</li> </ul>	Echantillon	#222222	#ffff	15.91:1 (succès au niveau AAA)	221 (succès)	663 (succès)
A	<ul style="list-style-type: none"> <li>HTML</li> <li>BODY</li> <li>Div#container</li> <li>UL#topnav</li> <li>LI</li> </ul>	Echantillon	#008000	#008000	5.14:1 (AAA succès pour le grand texte; AA pour le texte régulier)	179 (succès)	637 (succès)
A	<ul style="list-style-type: none"> <li>HTML</li> <li>BODY</li> <li>Div#container</li> <li>UL#topnav</li> <li>LI</li> </ul>	Echantillon	#ffff	#591789	11:1 (succès au niveau AAA)	199 (succès)	516 (succès)
A	<ul style="list-style-type: none"> <li>HTML</li> <li>BODY</li> <li>Div#container</li> <li>Div#utility</li> <li>Div#radio</li> </ul>	Echantillon	#000000	#ffff	21:1 (succès au niveau AAA)	255 (succès)	765 (succès)
A	<ul style="list-style-type: none"> <li>HTML</li> </ul>	Echantillon	#55009d	#ffff	11.21:1 (succès au niveau AAA)	211 (succès)	523 (succès)

L'avantage de cette application est de fournir un rapport détaillé des tests des couleurs et de fournir l'arborescence du code HTML testé ainsi que les couleurs de texte et de fond.

## 5) W3C markup validation service

W3C Markup Validation Service est un outil en ligne permettant de valider le code HTML et XHTML d'une page Web par rapport à la DTD définie dans le document. La page peut être transmise à l'outil à l'aide de son URI, en la chargeant sous forme de fichier ou en insérant directement son code source dans la zone de saisie de l'outil. La figure suivante montre l'interface de l'outil.

L'application génère un rapport recensant les erreurs de conformité à la DTD spécifiée. La figure suivante montre un exemple de rapport obtenu.

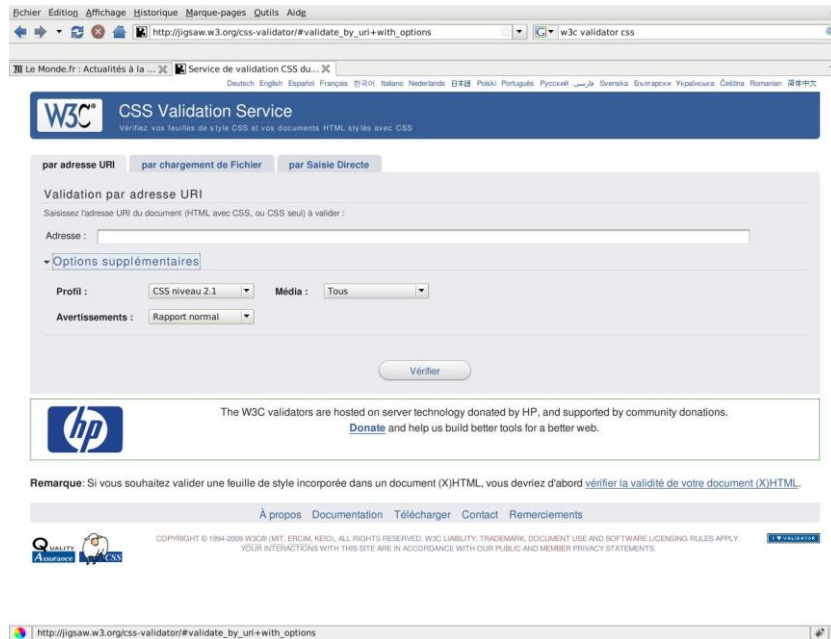


Le rapport détaille toutes les erreurs identifiées et donne les causes possibles des erreurs ainsi que la méthode permettant de les corriger.

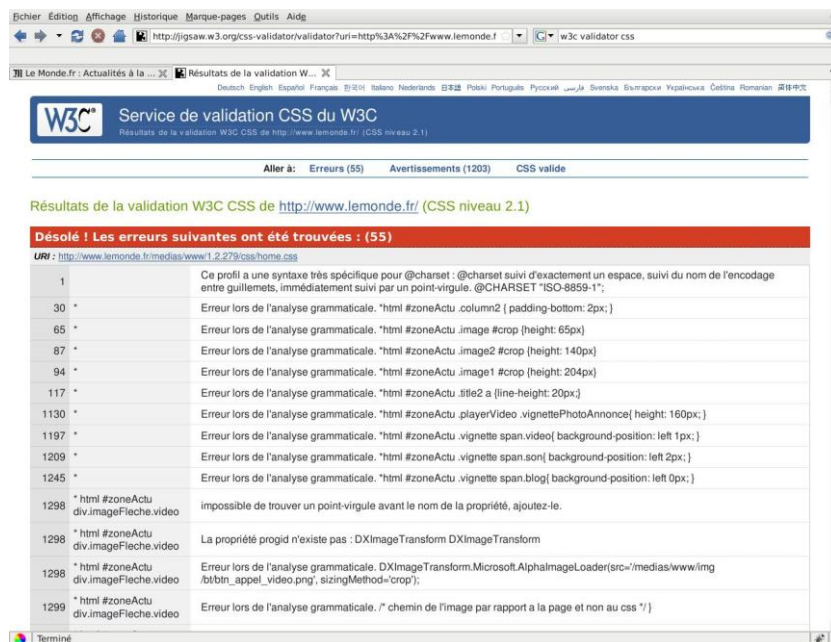
### 6) W3C Service de validation CSS

W3C Service de validation CSS est un outil en ligne permettant de valider le code CSS d'une feuille de style ou d'une page Web par rapport au niveau de CSS sélectionné. Le document peut être transmis à l'outil à l'aide de son URI, en la chargeant sous forme de fichier ou en insérant directement son code source dans la zone de saisie de l'outil. La figure suivante montre l'interface de l'outil.

# Annexes



L'application génère un rapport recensant les erreurs de conformité au niveau de CSS sélectionné. La figure suivante montre un exemple de rapport obtenu.



### 7.3.2. Outils testés et non retenus

#### 1) TAW3 (extension Firefox)

TAW existe sous forme d'extension Firefox. Il permet de valider la page en cours de consultation, d'un simple clic sur l'icône de l'extension, mais offre moins de fonctionnalités que la version installée.

Nous ne l'avons pas retenu car l'application ne permet pas de valider des pages non accessibles par Internet. En effet, l'analyse n'est pas exécutée par l'extension mais par un serveur qui consulte la page en ligne. Dans notre cas, la page n'étant pas accessible, le serveur ne peut pas analyser notre page.

#### 2) Web Accessibility Inspector (Version 5.11)

Web Accessibility Inspector est un outil de test développé par Fujitsu et permettant d'évaluer l'accessibilité d'un site Web vis-à-vis des recommandations du WCAG 1.0. La figure suivante nous montre l'interface graphique de l'application.



Il génère un rapport au format HTML recensant les erreurs de conformité découvertes, ainsi que les erreurs en feuilles de style. La figure suivante montre un exemple de rapport d'analyse produit par Web Accessibility Inspector.

**Fujitsu Web Accessibility Inspector 5.11**

**Diagnosis Result according to**  
**Web Content Accessibility Guidelines 1.0**  
 W3C Recommendation 5-May-1999  
 Copyright (c) 1999 W3C (MIT, INRIA, Keio), All Rights Reserved.  
 (http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/)

**Checked file**

File name	Number of problems
T:\Test Front - onglet.htm	154problem(s)
http://xtek.iut2.upmf-grenoble.fr/design_spe/webtek-98/css/index.css	32problem(s)
http://xtek.iut2.upmf-grenoble.fr/design_spe/webtek-98/css/default.css	77problem(s)

**Checked file (1)** Next file

**Target**

File name: T:\Test Front - onglet.htm  
 CSS file referred to: http://xtek.iut2.upmf-grenoble.fr/design\_spe/webtek-98/css/index.css, http://xtek.iut2.upmf-grenoble.fr/design\_spe/webtek-98/css/default.css

**Result**

Number of problems: 154 problem(s) have been found.

Problem detail  Source list

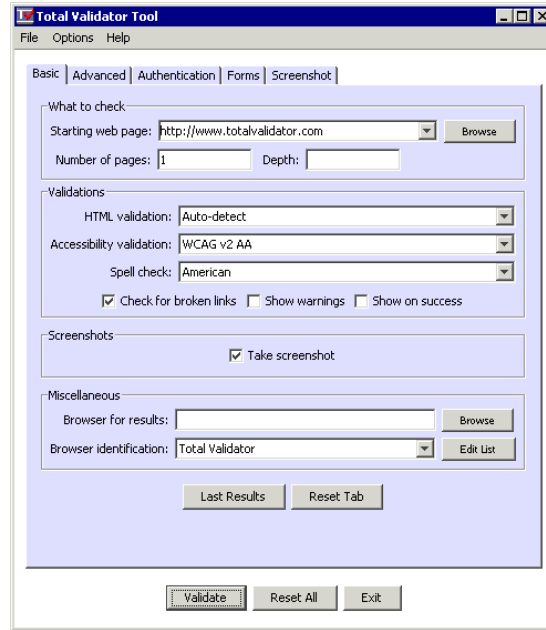
**Problem outline**

Priority level	Classification	Number of problems	
1	⊘	8problem(s)	⊘ This item requires some corrections.
	⚠	5problem(s)	
	⊘	3problem(s)	⚠ This item does not constitute a problem depending on its purpose. Review its purpose and correct it as necessary.

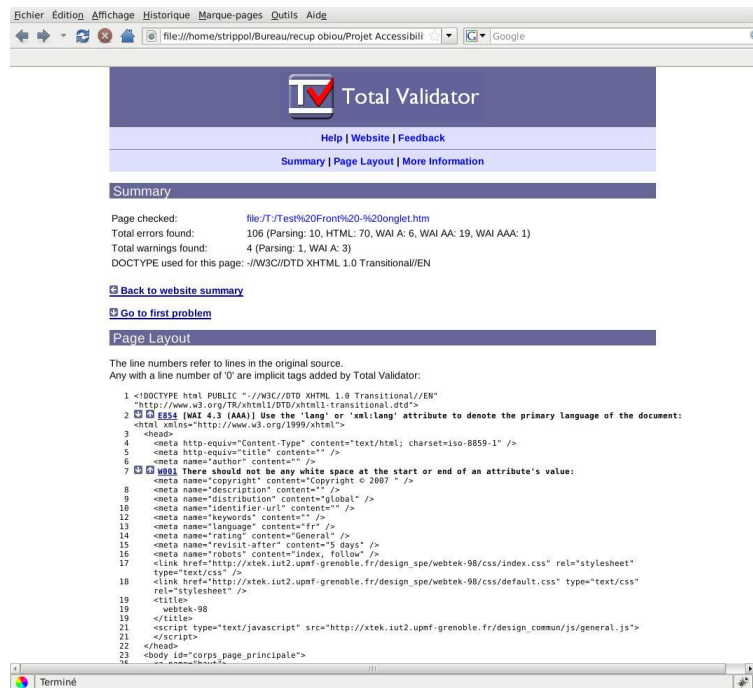
Cet outil n'a pas été retenu car il s'est avéré moins performant en analyse que l'application TAW3.

### 3) Total Validator (Version 5.4)

Total Validator est un outil permettant de tester la conformité d'un site Web vis-à-vis des recommandations du WCAG 1.0, ainsi que de valider le code HTML des pages. La figure suivante montre l'interface graphique de l'application.



L'application génère un rapport au format HTML recensant toutes les erreurs identifiées. La figure suivante montre un exemple de rapport obtenu.



L'application n'a pas été retenue car elle s'est retrouvée moins performante que l'application TAW3. Il est à noter que nous avons aussi testé l'outil Total Validator en tant qu'extension Firefox que nous avons écarté pour les mêmes raisons que l'extension Firefox de TAW.





# MEMOIRE D'INGENIEUR C.N.A.M. en INFORMATIQUE

## Mise en conformité d'un logiciel universitaire de publication de contenus Web à vocation publique aux normes d'accessibilité du Web

Jérôme STRIPPOLI

Grenoble, le 30 juin 2010

---

### Résumé

Dans le milieu des années 90, afin de permettre l'accès au Web à tous, le WAI, groupe du W3C, a défini dans le WCAG des directives expliquant comment rendre les contenus Web accessibles aux personnes handicapées. Ce document est devenu depuis une norme internationale sur laquelle reposent toutes les initiatives visant à rendre le Web accessible.

Depuis la parution en 2005 de la loi pour « l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées » et la publication de son décret d'application le 14 mai 2009, les sites à vocation publique doivent être accessibles aux personnes handicapées. Le RGAA 2.2, créé par la DGME et reposant sur le WCAG 2.0, sert de référentiel à cette mise en accessibilité.

X-TEK est un générateur de plates-formes développé par l'IUT2 qui a servi à générer bon nombre de sites institutionnels. Il rentre dans le cadre de cette loi et se doit de devenir accessible.

Pour rendre X-TEK accessible, nous avons réalisé un audit préalable de son accessibilité. Ensuite, nous avons procédé à sa mise en conformité vis-à-vis du référentiel RGAA 2.2.

**Mots-clés :** Accessibilité, Handicap, WCAG, RGAA, Web, X-TEK

---

### Abstract

In the mid-nineties, in order to give everyone the opportunity to use the Web, the WAI, W3C group, has defined the WCAG guidelines explaining how to make Web contents accessible to disabled people. This document has become an international standard on which are founded all initiatives allowing the Web accessibility.

Since the publication in 2005 of the law for "equality of rights and opportunities, the involvement and citizenship of disabled people" and the release of its implement decree on May 14, 2009, public purposes sites must be accessible to disabled people. The RGAA 2.2. created by the DGME and relying on the WCAG 2.0. will be the reference for this accessibility development.

X-TEK is a platform generator developed by the IUT 2 which has enabled the creation of many institutional sites. It stands in confines of this law and must be accessible.

To allow X-TEK accessibility, we start making a preliminary audit of its accessibility. Then, we shall study its compliance with the RGAA 2.2.

**Keywords :** Accessibility, Disability, WCAG, RGAA, Web, X-TEK