

UNIVERSITE CATHOLIQUE DE LOUVAIN



**LOUVAIN**  
School of Management

# Travail de fin d'études

---

Analyse comparative des techniques d'adaptation du front-end d'un web service en fonction des contraintes imposées par le contexte de l'utilisateur

**Pas Mathieu**

Promoteur : Prof. Jean Vanderdonckt

Année académique 2010 - 2011

## **AVANT-PROPOS**

Je tiens à remercier mon directeur, le professeur Jean Vanderdonckt pour m'avoir proposé ce sujet intéressant et m'avoir donné des directives précises.

Je tiens également à remercier mademoiselle Vivian Motti pour son aide et ses explications.

Enfin, un merci très particulier à Cindy Jacquemin pour sa relecture attentive et ses corrections pertinentes

## Table des matières

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	CONTEXTE GÉNÉRAL .....	1
1.2	ENONCÉ DE L'OBJECTIF CENTRAL.....	2
1.2.1	<i>Sur quoi allons-nous nous concentrer ?</i> .....	3
1.2.2	<i>Le contexte : une définition du W3C</i> .....	3
1.2.3	<i>Qu'entend-on par « adaptation » ?</i> .....	4
1.3	HYPOTHÈSES, MÉTHODES, PLAN DE LECTURE .....	5
<b>2</b>	<b>ETUDE CRITIQUE DE L'EXISTANT .....</b>	<b>8</b>
2.1	LIEU DU PROCESSUS D'ADAPTATION .....	8
2.1.1	<i>Server-side Adaptation</i> .....	8
2.1.2	<i>Client-side adaptation</i> .....	10
2.1.3	<i>Intermediate adaptation (Proxy)</i> .....	10
2.1.4	<i>Où réaliser ce processus d'adaptation ?</i> .....	12
2.2	CRITÈRES D'ÉVALUATION DES TECHNIQUES.....	12
2.3	SOLUTIONS PRATIQUES EXISTANTES .....	13
2.3.1	<i>Opéra mini</i> .....	14
2.3.2	<i>Skweezer</i> .....	15
2.3.3	<i>Google Mobilizer</i> .....	16
2.3.4	<i>BareSite</i> .....	17
2.3.5	<i>Mowser</i> .....	17
2.3.6	<i>Quels outils choisir ?</i> .....	18
2.4	TECHNIQUES THEORIQUES .....	18
2.5	STANDARDS WEB ET TECHNOLOGIES UTILISÉES PAR LES TECHNIQUES D'ADAPTATION .....	20
2.5.1	<i>CSS Media Queries</i> .....	20
2.6	PRÉSENTATION DES TECHNIQUES .....	24
2.6.1	<i>Style</i> .....	25
2.6.2	<i>Layout</i> .....	28
2.6.3	<i>Contenu</i> .....	29
2.6.4	<i>Structure</i> .....	30
2.6.5	<i>Navigation</i> .....	34
2.6.6	<i>Interaction</i> .....	35
2.6.7	<i>Techniques d'adaptation sous-jacentes</i> .....	36
2.6.8	<i>Tableau recapitulatif des techniques théoriques</i> .....	48
<b>3</b>	<b>PROPOSITION DE SOLUTION.....</b>	<b>49</b>

## II.

3.1.1	<i>La notion de pertinence : méthode de détermination.....</i>	49
3.1.2	<i>Principe général .....</i>	51
3.1.3	<i>Fonctionnement global .....</i>	51
3.1.4	<i>Conséquences.....</i>	53
<b>4</b>	<b>ETUDE DE CAS .....</b>	<b>55</b>
4.1	SITE DE PETITE TAILLE .....	55
4.1.1	<i>Opera Mini .....</i>	55
4.1.2	<i>Skweezer .....</i>	56
4.1.3	<i>Notre solution .....</i>	56
4.2	SITE DE TAILLE MOYENNE .....	57
4.2.1	<i>Opera Mini .....</i>	57
4.2.2	<i>Skweezer .....</i>	57
4.2.3	<i>Notre solution .....</i>	58
4.3	SITE DE TAILLE IMPORTANTE.....	58
4.3.1	<i>Opéra mini .....</i>	59
4.3.2	<i>Skweezer .....</i>	60
4.3.3	<i>Notre solution .....</i>	60
<b>5</b>	<b>CONCLUSION .....</b>	<b>61</b>
5.1	RÉSUMÉ .....	61
5.2	LIMITES .....	61
5.3	TRAVAUX FUTURS .....	61
<b>6</b>	<b>BIBLIOGRAPHIE .....</b>	<b>63</b>

# 1 Introduction

## 1.1 Contexte général

A l'heure d'une lente mais certaine transition vers un Internet IPv6, il est nécessaire de dresser un constat. En effet les quelques  $2^{32}$  (soit plus de 4 milliards) d'adresses IP disponibles sont sur la voie de la pénurie d'ici quelques années. Bien que cela puisse paraître étonnant à première vue, l'explosion du nombre d'utilisateurs dans les pays à haute densité démographique comme l'Inde ou la Chine peut en partie expliquer le phénomène. La multiplication du nombre d'adresses employées par un individu peut également constituer un facteur significatif expliquant ce phénomène. Smartphones, PC, GSM et plus récemment les tablettes, pc ultraportables, lecteurs MP3 et consoles de jeux, sont autant de dispositifs capables d'accéder au réseau global, nécessitant l'allocation d'une adresse IP publique qui leur est propre. Un même utilisateur peut donc consommer plus de 3 adresses simultanément, un cas de figure que n'avaient certainement pas prévu les concepteurs du célèbre protocole.

Outre un problème général d'adressage, la multiplication de ces appareils soulève la question de l'accessibilité d'Internet. Les enjeux proviennent du caractère hétérogène de ces moyens d'accès qui possèdent tous des caractéristiques qui leur sont propres : autonomie, taille du dispositif, capacités de calcul, espace mémoire disponible, formats supportés, résolution de l'écran, vitesse de connexion. Autant de paramètres qui rentrent en ligne de mire lors de la consultation d'une page web qui reste la même, peu importe le moyen utilisé pour y accéder.

Cet engouement pour les tablettes et autres Smartphones est relativement récent et l'on pourrait croire qu'il s'agit d'une mode passagère, d'une tendance générale pour les nouvelles technologies qui va s'estomper dans un avenir proche. Après tout, un PC même portable est bien plus complet, puissant et les applications Web sont conçues et optimisées pour être affichées sur un tel support. Cependant cette évolution risque de ne pas s'arrêter, comme le montre la Figure 1, sur laquelle on peut voir une courbe de progression des pages web consultées via le navigateur pour mobiles « Opéra mini », entre juin 2009 et mai 2011. Bien que cela ne représente qu'une partie de la navigation mobile, ces données nous permettent d'avoir un aperçu de la réalité, et l'on ne peut que constater la rapidité de la progression.

## Pages transcoded

Opera Mini users viewed over 63.3 billion pages in May 2011. Since April, page views have gone up 9.4%. Since May 2010, page views have increased 123.8%.

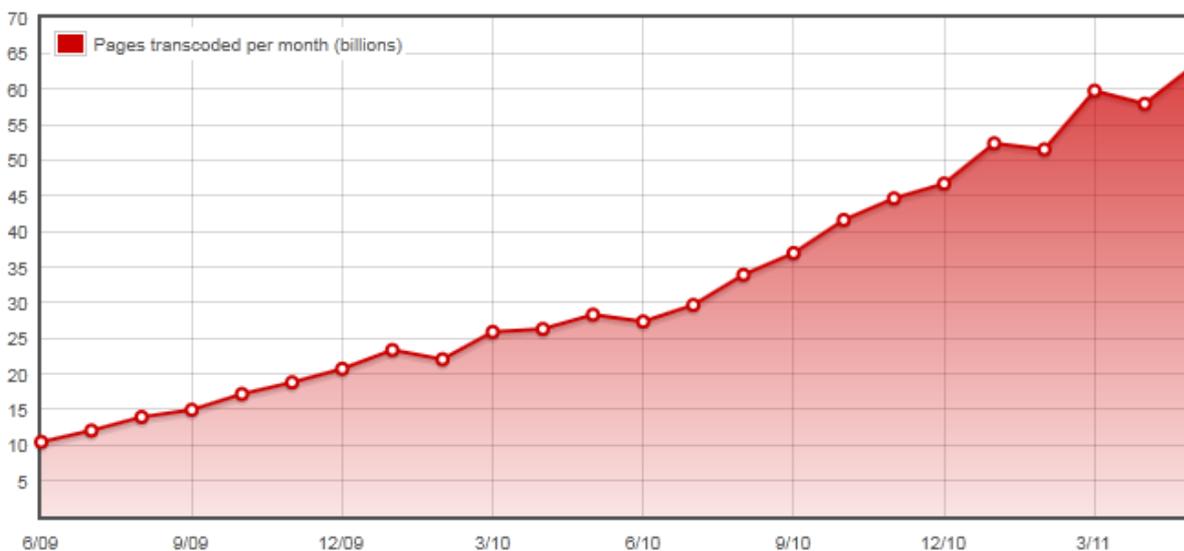


Figure 1- Évolution du nombre de pages traitées par « Opéra mini », un navigateur optimisé pour mobiles<sup>1</sup>

## 1.2 Énoncé de l'objectif central

Cette nouvelle réalité numérique nous force à penser un Internet nouveau, accessible presque partout et par un grand nombre de dispositifs différents. De nouveaux challenges sont apparus et les concepteurs d'applications Web devront de plus en plus se pencher sur la question de l'accessibilité pour rester compétitifs. Or, dans cette période « charnière » actuelle, les sites proposant une version mobile sont encore trop rares ou ne répondent pas exactement aux besoins et spécificités des différents médias. Pourtant, la demande de contenu optimisé est bien réelle et des solutions doivent être mises en place.

L'adaptation des documents Web permet de répondre à cette demande, dans l'attente d'une prise de conscience globale des webmasters. Des recherches ont débuté depuis quelques années déjà et tentent de traiter le nombre incalculable d'aspects différents relatifs à cette discipline. Ce travail a pour objectif de mettre au point une solution viable basée sur ces techniques et qui permettrait d'adapter une page Web en fonction de son contexte d'utilisation.

<sup>1</sup> Source : <http://www.opera.com/smw/2011/05/>

### 1.2.1 Sur quoi allons-nous nous concentrer ?

Comme le titre du présent travail le mentionne, nous allons nous concentrer sur l'adaptation du front-end d'un web service en fonction du contexte dans lequel est plongé l'utilisateur. Ces termes méritent quelques explications.

Le front-end est un terme abondamment utilisé dans différents domaines et possède même plusieurs significations en informatique en fonction du contexte dans lequel il est employé. Dans le cas d'applications logicielles, le front-end correspond à l'entité qui sert de point d'entrée à l'utilisateur pour y encoder diverses données. En d'autres termes, on peut affirmer qu'il s'agit de l'interface graphique (GUI) via laquelle l'utilisateur interagit avec le système.

Par web service, nous entendons non pas le mécanisme de communication du même nom, mais bien un site Internet ou application web.

Enfin les notions de contexte et d'adaptation méritent d'être détaillées dans les sections qui suivent.

### 1.2.2 Le contexte : une définition du W3C

Pour appréhender ce concept nous partirons des travaux du World Wide Web Consortium (W3C), le célèbre organisme de standardisation. Face à la multiplication des moyens d'accès à Internet, des groupes de travail se sont formés pour répondre aux nouveaux challenges. Leurs résultats et travaux en cours sont repris sur le site officiel du consortium, dans la section « *Web of devices* »<sup>2</sup>. Nous nous intéresserons particulièrement à la sous-section « *Device Independence and Content Adaptation* » qui se concentre sur l'adaptation du contenu en fonction du contexte, dans l'optique globale « d'avoir un web qui soit accessible à partir de n'importe quel type d'appareil »<sup>3</sup>.

Le contexte (appelé *delivery context*, soit contexte de remise) se définit comme : « *Un ensemble d'attributs qui caractérise les capacités du mécanisme d'accès, les préférences de l'utilisateur et d'autres aspects du contexte dans lequel une page web doit être délivrée.* » (1)

Le mécanisme d'accès est « une combinaison de matériels (ordinateurs personnels, Smartphones, tablettes, ...) et de programmes (navigateurs etc.) qui permettent à l'utilisateur d'accéder à une page web.

---

<sup>2</sup> <http://www.w3.org/standards/webofdevices/>

<sup>3</sup> <http://www.w3.org/standards/webofdevices/independence>

Il s'agit donc d'un environnement très hétérogène qui possède de nombreuses caractéristiques variables, parmi lesquelles on peut citer de manière non exhaustive :

Catégorie	Caractéristique
<b>Interaction</b>	<ul style="list-style-type: none"> <li>- Modalités de présentation et leurs paramètres (output)</li> <li>- Modalités de capture et leurs paramètres (input)</li> </ul>
<b>Connexion</b>	<ul style="list-style-type: none"> <li>- Bande passante disponible</li> <li>- Protocoles acceptés</li> </ul>
<b>Capacité de l'agent utilisateur</b>	<ul style="list-style-type: none"> <li>- Capacités de traitement</li> <li>- Formats acceptés (types MIME etc.)</li> </ul>

Pour une liste plus complète, voir (1)

### 1.2.3 Qu'entend-on par « adaptation » ?

On peut, à première vue, se demander pourquoi clarifier cette notion qui, dans la vie de tous les jours, est sans équivoque est simple à cerner. On pense notamment à l'entreprise qui s'adapte à son environnement, aux capacités d'adaptation de certains animaux, ou encore au fait de s'exprimer de façon adaptée à un certain contexte, etc. De manière générale, on peut dire qu'il s'agit de prendre des dispositions face à un ensemble de contraintes sur lesquelles nous n'avons aucun pouvoir, aucune emprise.

Ce principe peut évidemment être transposé au monde de l'informatique, même si sa définition devient par là même moins évidente. De nombreuses recherches ont déjà été menées sur ce concept, engendrant une taxonomie conséquente et un large ensemble de définitions différentes, variant selon l'approche utilisée. Pour la suite du présent travail, nous tenons à préciser et clarifier la version de la notion d'adaptation que l'on utilise.

Dans notre approche centrée sur l'interface graphique ou « front-end », l'adaptation se définit comme un processus consistant à modifier toute ou partie de cette interface dans le but de répondre aux besoins de l'utilisateur. De manière non exhaustive, il peut s'agir de supprimer, masquer, déplacer, tronquer, redimensionner certains éléments. On distingue deux sous-processus dans l'adaptation, selon le rôle joué par le système ou l'utilisateur lui-même : l'adaptabilité et l'adaptativité (ce dernier terme qui n'est pas référencé par les dictionnaires peut être vu comme une traduction littérale approximative de l'anglais « *adaptivity* »). (2)

L'adaptabilité désigne la capacité de l'utilisateur à adapter l'interface graphique selon ses propres besoins, ses préférences. On peut également voir ça du point de vue du système qui doit dès lors être capable de s'adapter en fonction de ce que désire l'utilisateur. L'adaptabilité fait donc intervenir deux acteurs, l'utilisateur qui définit explicitement ses préférences et le système qui doit pouvoir y répondre. On songe par exemple à l'utilisateur qui définit un ordre dans lequel il préfère voir apparaître différentes sections d'une page web.

L'adaptativité désigne, quant à elle, la capacité du système à adapter le front-end aux besoins de l'utilisateur sans intervention préalable de ce dernier. Le processus est plus dynamique puisque le système doit au préalable déterminer à quelles contraintes il doit s'adapter. Il peut par exemple s'agir de modifier l'apparence d'une page web en fonction de la taille de l'écran sur lequel elle est visionnée.

Enfin, le W3C en donne une définition particulièrement concrète et pertinente pour notre analyse : « *A process of selection, generation or modification that produces one or more perceivable units in response to a requested uniform resource identifier in a given delivery context* » (3)

Il s'agit donc d'un processus, ce qui sous-entend un traitement, un certain nombre d'opérations nécessaires à réaliser sur un document pour produire le résultat final. Les opérations réalisables sont nombreuses et seront détaillées plus loin dans ce travail.

Ce processus produit une ou plusieurs unités perceptibles, c'est-à-dire des unités de données qui peuvent être traitées par le dispositif de destination pour être présentées d'une manière appropriée à l'utilisateur final et permet à ce dernier d'interagir avec le contenu délivré. Cela doit être rendu possible pour toute requête réalisée vers une même URL dans un contexte donné.

### **1.3 Hypothèses, méthodes, plan de lecture**

Pour que nos résultats soient valables et consistants, il nous est nécessaire de poser des hypothèses qui délimitent notre champ d'action. Premièrement, nous concentrerons notre étude sur un sous-ensemble des techniques d'adaptation existantes. En effet, certaines méthodes sont trop spécifiques ou demandent des connaissances pointues dans un domaine donné. C'est le cas notamment pour des traitements avancés sur les images, la vidéo ou

6.

encore le son. Nous nous devons en effet d'instaurer un degré d'abstraction, pour se concentrer uniquement sur notre objectif et ne pas se perdre dans des analyses techniques approfondies qui sortent du cadre de ce travail et pourraient, à elles seules, faire l'objet d'une étude.

La deuxième hypothèse est que nous partons du principe que toutes les techniques passées en revue fonctionnent, peuvent être combinées entre elles lorsque cela est possible, produisent un résultat correct, ne provoquent pas d'erreur et ne créent pas un sentiment de confusion chez l'utilisateur. Les solutions ne seront pas concrètement implémentées mais une simulation sera effectuée en fonction des spécifications fournies.

Pour réaliser notre analyse, nous suivrons une méthodologie consistant à partir des bases et principes théoriques définissant les techniques, pour ensuite les appliquer à des cas concrets afin de les comparer et d'en tirer les principaux avantages et inconvénients.

Nous commencerons, dans une première partie, par énoncer les principes généraux d'adaptation, pour ensuite décrire les solutions déjà existantes sur le marché. Chaque solution sera testée sur un exemple concret de site web, avant d'être évaluée et d'en définir les principaux avantages et inconvénients. Nous prendrons ensuite un point de vue plus théorique, en explorant un document du W3C qui contient de nombreuses techniques d'adaptation. Nous décrirons chaque méthode systématiquement, à l'aide d'un gabarit spécifique.

Dans la seconde partie, nous proposerons une solution alternative, qui se basera sur les techniques d'adaptation classiques. Cette solution consiste à tenir compte des tendances de navigation des utilisateurs pour mettre en évidence des parties du contenu jugées pertinentes.

La troisième partie sera véritablement le laboratoire de ce travail, puisque c'est celle qui accueillera les simulations concrètes. Nous testerons la solution préalablement établie sur des sites Internet réels de différentes envergures : petit, moyen et grand. Cela nous permettra de comparer notre solution « maison » avec les solutions pratiques qui sont actuellement utilisées.

Nous terminerons naturellement par tirer les conclusions de ce que nous avons réalisé. Notre solution est-elle satisfaisante ? Qu'avons-nous pu observer ? Quelles sont les limites et les perspectives d'avenir pour l'adaptation ? Nous tenterons de donner une réponse à chacune de ces questions.

## 2 Etude critique de l'existant

Avant de détailler différentes techniques d'adaptation et de les combiner pour obtenir une proposition viable, il est nécessaire d'étudier préalablement le paysage des solutions existantes. De nombreuses compagnies ont confectionné leur propre outil d'adaptation, avec plus ou moins de succès et de popularité. Chaque solution sera présentée au moyen d'une description, accompagnée d'un exemple d'application réel et, enfin, des principaux avantages et inconvénients. De cette manière, nous pourrions sélectionner la ou les meilleures solutions existantes et les comparer avec notre proposition, dans la section qui suit.

### 2.1 Lieu du processus d'adaptation

Peu importe les mécanismes internes utilisés, une question importante à se poser lorsque nous sommes amenés à développer un système d'adaptation pour les pages web est de savoir *où* aura lieu le processus d'adaptation. Dans l'architecture client-serveur actuelle utilisée sur Internet, deux possibilités extrêmes s'offrent à nous : exécuter le processus soit du côté du client, soit du côté du serveur. Nous verrons, par la suite, qu'un troisième acteur peut intervenir et effectuer l'adaptation entre les deux parties principales.

#### 2.1.1 Server-side Adaptation

Il s'agit de l'adaptation réalisée du côté du serveur dont le principe est illustré ci-dessous. Le serveur traite toute requête qui lui est destinée (ex : demande d'une URL particulière) et tente de récupérer un maximum d'informations sur le *delivery context*, afin de dresser le plus fidèlement possible le « portrait » du dispositif de destination. Ces informations lui permettront de sélectionner les ressources les plus adaptées dont il dispose pour présenter une page web dans un format adéquat, ou d'effectuer des opérations sur mesure pour réajuster le contenu initial et fournir un contenu adapté.

Le nuage World Wide Web peut recouvrir l'ensemble des dispositifs existants disposant d'un accès Internet. Le nombre de cas différents à traiter est donc généralement très large et il est nécessaire de prévoir des alternatives pour chaque catégorie d'appareils.

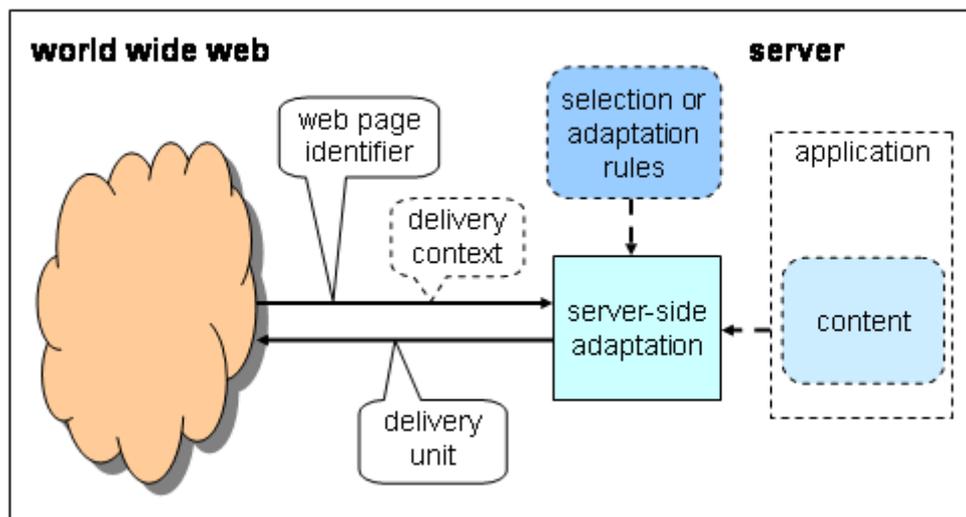


Figure 2 - Server-side adaptation (3)

Le principal avantage de cette approche est que tout le processus est géré par le serveur distant, ce qui épargne au client un travail plus que fastidieux qu'il est susceptible de ne pas pouvoir réaliser et dans lequel il risque de consommer de grande quantité d'énergie (batterie limitée). Il y a bien sûr d'autres avantages au fait de recevoir un contenu adapté :

- La taille occupée par le contenu est réduite, l'information est donc transmise plus rapidement, ce qui est plus que préférable dans un environnement où les vitesses de connexion peuvent être très lentes (manque de couverture, hardware limité). Cela évite également de dépasser un quota de données transférées et réduit le temps de réponse, choses importantes dans un environnement où les fournisseurs d'accès facturent à un prix élevé ce type de communication (3G, GPRS, etc.)
- Cette taille réduite évite également de surcharger la mémoire du dispositif destinataire. Les GSM et autres systèmes embarqués peuvent présenter des faiblesses à ce niveau-là.
- Le contenu présenté peut être simplifié (simplification des polices, transformation de formats spéciaux en équivalents standards). Certains formats de fichiers et autres polices spécifiques ne sont pas pris en compte par tous les appareils.
- Le niveau de sécurité est élevé car la connexion client-serveur est directe

Malgré ces nombreux avantages, cette solution comporte un inconvénient. En effet, le serveur dispose de grandes capacités de calcul, mais ce n'est pas son rôle de réaliser des transformations sur du contenu. Le rôle principal du serveur est de répondre aux requêtes qui lui sont adressées et cette « charge » supplémentaire risque de limiter son activité en cas d'affluence importante d'utilisateurs mobiles et classiques.

Une alternative pour pallier à ce problème est d'inclure l'auteur de l'application Web dans le processus d'adaptation. Plutôt que de tout automatiser, ce dernier peut créer une version pour mobiles, relativement générale, permettant au serveur de la délivrer si certaines conditions sont satisfaites, sans que de gros calculs soient effectués. Cette idée est de plus en plus présente dans l'esprit des développeurs Web mais, à l'heure actuelle, on constate que ce sont principalement les sites d'importantes multinationales (Google, Amazon, Wikipedia etc.) qui ont franchi le pas et bénéficient d'une version optimisée de leur site pour mobiles.

### 2.1.2 Client-side adaptation

A l'opposé de l'adaptation par le serveur, on retrouve le même processus mais cette fois-ci réalisé par le client. Cette solution est évidemment moins performante et nettement moins recommandée que la précédente pour toutes les raisons que l'on vient de citer (le principal facteur étant les capacités limitées des dispositifs de petite taille).

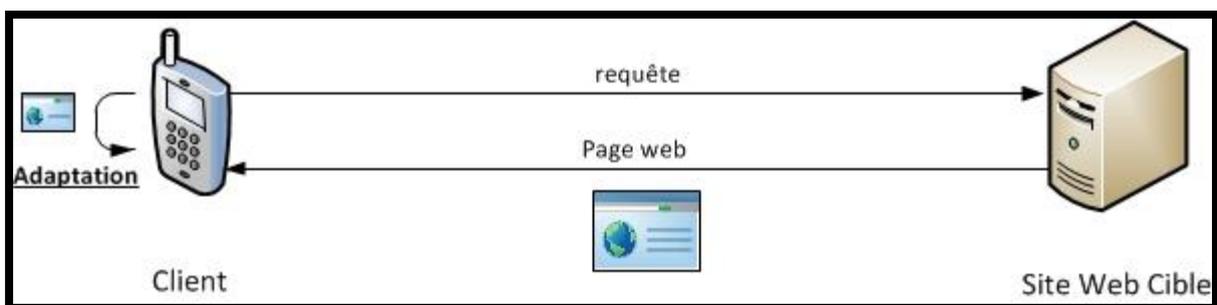


Figure 3 - Client-side adaptation

Néanmoins c'est une solution qui peut être envisagée pour des tâches d'adaptation mineures (substitution des polices, par exemple.). Le principal avantage étant le niveau de sécurité offert par la connexion directe avec le serveur distant.

### 2.1.3 Intermediate adaptation (Proxy)

Entre ces deux extrêmes, nous trouvons « l'intermediate adaptation », via l'utilisation d'un proxy, illustrée par la figure ci-dessous.

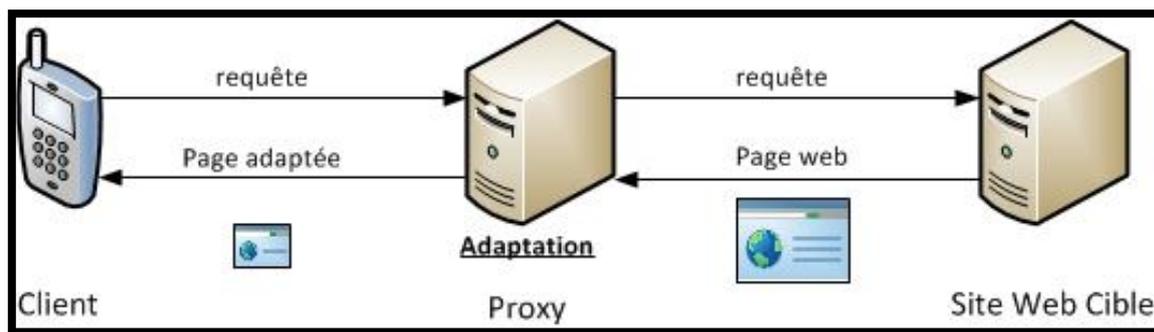


Figure 4 – Intermediate Adaptation

Prenons un exemple pour expliquer son principe de fonctionnement. Imaginons le cas d'un utilisateur (le client), qui désire accéder à un site Internet en utilisant son Smartphone ou un autre terminal de taille réduite. Pour recevoir une version adaptée de la page web, il envoie sa requête au proxy, qui se chargera également du processus d'adaptation. Ce dernier transfère la requête initiale au site web cible et récupère, en réponse, la page originale générée. Le plus souvent, cette page est optimisée pour être affichée par des ordinateurs conventionnels et possède un contenu qui peut être riche, afficher des éléments de grandes tailles et de nombreuses couleurs, etc. Le rôle du transcodeur (proxy), dans ce schéma, est d'adapter cette page aux contraintes imposées par l'utilisation du Smartphone (ex : La taille, la bande passante, la puissance, le nombre de couleurs et de formats supportés sont potentiellement réduits). Des algorithmes sont donc exécutés au niveau du transcodeur, manipulant ainsi le document et permettant de délivrer une page Web plus légère et plus facile à lire pour le dispositif de destination.

Bien que rajoutant un acteur supplémentaire dans le chemin des données, cette solution offre d'importants avantages regroupant les meilleurs points des deux techniques précédentes.

- Le serveur est totalement libéré de la charge supplémentaire que représentait l'adaptation. Il délivre les pages dans leur format original, comme si le destinataire était un ordinateur classique. La transparence est totale.
- Le client reçoit la version adaptée du contenu qui est optimisée, allégée et utilisable par l'utilisateur
- Le proxy est généralement un serveur dédié à cette pratique. Il est hébergé par une société qui a investi du temps, de l'argent et qui a développé des techniques élaborées pour rendre des contenus adaptés de haute qualité.

- Des mécanismes de mise en cache peuvent être mis en place.

Toute solution n'a pas que des avantages et celle-ci n'échappe pas à la règle. Les inconvénients suivants peuvent être constatés :

- Le proxy ajoute un niveau d'indirection sur le chemin des données. Cela peut augmenter le temps de réponse puisque plusieurs requêtes sont générées.
- Au niveau de la sécurité, l'ajout d'un maillon dans la chaîne réduit le niveau global et l'utilisateur peut être méfiant à l'idée de confier ses identifiants personnels de connexion à un site tiers.
- Vu que le serveur appartient à une société : s'il est indisponible ou surchargé, le site cible ne sera pas consultable

#### **2.1.4 Où réaliser ce processus d'adaptation ?**

Les trois alternatives offrent chacune leur lot d'avantages et inconvénients. En écartant l'adaptation côté client, qui ne peut pas traiter la totalité du processus, et en écartant également le fait que l'auteur ait optimisé son application Web pour un rendu mobile, les deux meilleures solutions sont donc d'adapter le contenu sur le serveur ou sur le proxy.

Dans la pratique, la solution du proxy est la plus utilisée. Cela s'explique notamment par le fait que l'on ne peut pas imposer aux utilisateurs d'installer des solutions sur leur dispositif, ni aux webmasters de configurer ou d'installer des outils sur leur serveur. Au final, les sociétés éditrices ont un périmètre d'action réduit et ne peuvent réellement influencer que sur leurs propres serveurs.

Une solution qui pourrait également être envisagée consisterait à répartir le processus d'adaptation sur tous les maillons de la chaîne (3 entités sont représentées sur le schéma, mais il peut y en avoir d'autres. Le serveur peut faire appel à des entités tierces pour effectuer certaines opérations d'adaptation, plusieurs proxys peuvent être présents entre le client et le serveur, etc.).

## **2.2 Critères d'évaluation des techniques**

Afin d'évaluer les différentes techniques, il est nécessaire de définir les critères sur lesquels nous allons nous baser pour tirer nos conclusions. Pour cela, il est indispensable de se

placer au niveau de l'utilisateur final, qui tente d'accéder à une page donnée. Nous pouvons dégager ces trois critères intéressants :

1. Le contenu adapté est fonctionnel : L'utilisateur peut profiter des fonctionnalités du site, naviguer de pages en pages, accéder à tout contenu (images, vidéos, textes) présent sur une page originale, utiliser les formulaires etc.
2. Le poids du contenu adapté : La taille du document délivré doit être réduite par rapport au contenu original, pour les raisons évidentes déjà énoncées. Cependant, il y a un équilibre à trouver entre espace mémoire occupé et qualité de présentation. Les images, sons et vidéos représentent la plus grosse part du poids total d'une page Web. La réduction de leur taille et/ou qualité permet de réduire de manière importante ce poids, mais peut également rendre ces ressources inutilisables si elle est poussée à l'extrême (images trop petites/illisible, bande son dégradée, vidéo pixélisée...)
3. Le contenu présenté est clair et ne sème pas la confusion chez l'utilisateur : L'utilisateur ne doit pas être perdu sur la version adaptée du contenu, la « mise en page » doit être correcte et empêcher que certains éléments se chevauchent, etc. Les solutions mettant en œuvre une scroll bar horizontale, par exemple, sont à éviter.

### 2.3 Solutions pratiques existantes

La majorité des solutions existantes se présente sous forme de proxy. Nous avons rassemblé quelques une de ces solutions gratuites, des plus populaires mais également des moins connues, afin d'avoir un large paysage à faire passer sur le banc de test. Chaque approche sera évaluée sur le résultat qu'elle produit. Pour effectuer cette analyse, nous avons choisi comme site « cobaye » le site de l'UCL<sup>4</sup>, et plus particulièrement sa page d'accueil. Cette page est intéressante puisqu'elle combine plusieurs aspects qui ne la rendent pas si simple que ça à traiter :

- Elle se décompose en plusieurs parties : deux menus latéraux qui entourent une zone de contenu, au centre, qui est elle-même divisée.
- Elle contient des images de différentes tailles
- De nombreux liens sont présents partout sur la page, y compris au-dessus de la bannière

---

<sup>4</sup> [www.uclouvain.be](http://www.uclouvain.be)



Figure 5 - Page d'accueil du site de l'UCL

### 2.3.1 Opéra mini

« Opéra mini » est le nom de la version mobile du célèbre navigateur. Initiée en 2005 cette solution peut être installée sur un très grands nombre de dispositifs (dont les tablettes), disposant pour la plupart d'une version optimisée du logiciel<sup>5</sup>. Le résultat d'adaptation est montré sur les deux images pertinentes que nous avons sélectionnées, voir ci-dessous.



<sup>5</sup> <http://www.opera.com/mobile/download/pc/>

Selon ces images, nous pouvons dresser un tableau reprenant les principaux atouts et inconvénients.

Avantages	Inconvénients
<ul style="list-style-type: none"> <li>- Le style et les couleurs sont conservés</li> <li>- Il n'y a pas de scroll bar horizontale</li> <li>- La page retournée ne provoque pas de sentiment de confusion chez l'utilisateur.</li> </ul>	<ul style="list-style-type: none"> <li>- Nécessite une installation de la part de l'utilisateur</li> <li>- Les images sont fortement compressées, rendant illisibles certaines informations</li> </ul>

### 2.3.2 Skweezer

Skweezer<sup>6</sup> est l'un des outils gratuits les plus populaires dans l'adaptation de contenu pour mobile. Lancé en 2003, il semble être l'un des acteurs les plus sérieux du marché. Après soumission de la page d'accueil de l'UCL, le site nous retourne le résultat suivant :



La première chose que l'on peut constater est que les informations de styles (polices, couleurs) sont perdues et ont été remplacées par un style uniforme : les liens sont colorés en vert et le texte normal est simplement affiché en noir sur fond blanc. Du point de vue des images, l'image de grande taille n'a pas été redimensionnée et sème la confusion (on ne voit pas tout le contenu) tout en augmentant le poids de la page.

Avantages	Inconvénients
<ul style="list-style-type: none"> <li>- Pas de scroll bar horizontale</li> <li>- Les informations textuelles sont lisibles et correctement agencées</li> </ul>	<ul style="list-style-type: none"> <li>- Image non redimensionnée</li> <li>- Perte d'informations de style</li> <li>- L'utilisateur doit se familiariser avec le style</li> </ul>

<sup>6</sup> <http://skweezer.com/>

- L'espace n'est pas gaspillé	propre de Skweezer
-------------------------------	--------------------

### 2.3.3 Google Mobilizer

On ne compte plus les outils que Google a développés dans de nombreux domaines. Les développeurs se sont cette fois confrontés au challenge de l'adaptation et nous proposent un outil « Google Mobilizer <sup>7</sup> » fonctionnant comme les autres sur le principe de proxy.



Avantages	Inconvénients
<ul style="list-style-type: none"> <li>- Pas de scroll bar horizontale</li> <li>- Les couleurs des polices correspondent au style original</li> <li>- L'utilisateur n'est pas totalement dépaycé</li> <li>- La structure du site est respectée</li> </ul>	<ul style="list-style-type: none"> <li>- Image non redimensionnée</li> <li>- Perte d'une partie des informations stylistiques</li> <li>- Espace très mal utilisé. De grands écarts entre les différentes parties auraient du être réduits au maximum</li> </ul>

<sup>7</sup> <http://google.com/gwt/n>

### 2.3.4 BareSite

BareSite<sup>8</sup> est un autre outil gratuit qui nous permet de naviguer via un dispositif à écran réduit sur Internet. Le contenu adapté est le suivant :



Avantages	Inconvénients
<ul style="list-style-type: none"> <li>- Pas de scroll bar horizontale</li> </ul>	<ul style="list-style-type: none"> <li>- Image non redimensionnée ni compressée</li> <li>- Perte des informations de style</li> <li>- Structure non respecté (cf. 1<sup>re</sup> image)</li> <li>- Accents non pris en charge</li> </ul>

### 2.3.5 Mowser

Mowser<sup>9</sup> est un outil de la société dotMobi qui propose un service similaire.



<sup>8</sup> <http://www.baresite.com>

<sup>9</sup> <http://mowser.com/>

Avantages	Inconvénients
<ul style="list-style-type: none"> <li>- Très bonne lisibilité</li> <li>- Structure conservée</li> <li>- La grande image est redimensionnée</li> </ul>	<ul style="list-style-type: none"> <li>- Les informations stylistiques sont perdues</li> <li>- Certains caractères ne sont pas pris en charge</li> <li>- Présence d'une scroll bar horizontale</li> </ul>

### 2.3.6 Quels outils choisir ?

Les outils présentés ne représentent qu'un aperçu de tous les outils existants, mais les meilleurs font partie de notre sélection. En comparant les différents résultats, on se rend compte que l'adaptation n'est pas une science exacte et que tous les outils ont leurs avantages et leurs inconvénients.

Pour la suite de ce travail, nous sélectionnerons deux outils avec lesquels nous comparerons les résultats de notre solution. Ces deux outils sont « Opera mini » et « Skweezer ». Ce choix nous paraît justifié dans la mesure où ces deux solutions présentent les meilleurs résultats. Opera est largement au-dessus des autres en proposant une version très fidèle à la version originale et qui est parfaitement utilisable sur un écran réduit. Skweezer quant à lui propose une version plus épurée mais totalement fonctionnelle et réfléchie.

Les autres solutions possèdent toutes des inconvénients qui nous semblent réduire drastiquement l'expérience de l'utilisateur : non optimisation de l'espace, contenu déstructuré, caractères illisibles.

## 2.4 Techniques théoriques

Il existe un grand nombre de techniques permettant de modifier et d'adapter un contenu quelconque à un contexte particulier. Ces techniques peuvent être classées dans différentes catégories, non exclusives, selon leur approche et l'aspect particulier qu'elles visent.

Le World Wide Web Consortium énumère une taxonomie de six catégories dans lesquelles peuvent être positionnées les différentes techniques (3).

- **Le style**

Sont concernés ici tous les éléments qui définissent l'apparence du texte. Nous pouvons citer, de manière non exhaustive, des attributs tels la couleur, la police de

caractères utilisée, l'interligne, la justification ou encore la taille des caractères. Il n'est donc pas question de modifier le contenu du texte mais bien sa présentation (ex : masquer certains blocs d'informations).

- **Le Layout**

Il s'agit de la manière dont sont mis en page les différents éléments constitutifs du contenu sur la page Web. Nous sommes ici dans un contexte visuel. Le layout sera directement impacté par une réduction de la taille de l'écran sur lequel le site est consulté. Il est donc question de modifier l'agencement, la taille ou l'apparence de blocs de contenu et non pas le contenu lui-même.

- **Le contenu**

Le contenu représente l'ensemble des informations présentes sur la page Web. Il s'agit véritablement du cœur d'un site Internet, la raison pour laquelle les utilisateurs viennent le visiter ou non. Le contenu constitue une véritable valeur ajoutée et peut se traduire sous forme de textes, images, vidéos, sons, voire une combinaison de ces éléments. Adapter du contenu consistera souvent à opérer une sélection sur ce qui doit effectivement être présenté en priorité (par exemple, pour un appareil de petite taille visitant le site d'un journal, présenter une liste des titres principaux au lieu d'un article détaillé est une solution pertinente).

- **La structure**

La structure représente les liens qui unissent les différentes parties du contenu d'un site. Prenons comme document une page HTML, elle possède une structure qui lui est propre. Scinder cette page en deux parties aura un impact sur sa structure, mais également sur la navigation (ajout de liens pour naviguer d'une partie scindée vers l'autre, etc.)

- **La navigation**

Regroupe l'ensemble des composants permettant à l'utilisateur de parcourir le site de façon rapide et simple. Nous pouvons penser aux éléments de menus et sous-menu qui constituent des raccourcis rapides vers des sections précises, des numéros de pages, des liens « suivant » et « précédent » pour parcourir un large contenu de manière séquentielle etc.

### - **L'interaction de l'application**

Dans l'ère actuelle d'un Web participatif, les utilisateurs ne se contentent plus de consulter un site mais veulent également émettre leur opinion, poser des questions, en un mot : s'exprimer. Ils utilisent à cet effet des moyens leur permettant de communiquer des informations au site comme les formulaires. Étant donné la taille importante de certains formulaires, nous étudierons des techniques permettant de réduire cette taille tout en gardant les fonctionnalités proposées. Seule la partie présentation sera modifiée.

## **2.5 Standards Web et technologies utilisées par les techniques d'adaptation**

Avant de décrire de manière détaillée chaque technique d'adaptation, il est nécessaire de parler des standards ou travaux du W3C qui permettent de réaliser de manière opérationnelle le but poursuivi. Avec la standardisation croissante du Web, on réalise rapidement l'avantage qui découle de l'utilisation de ces recommandations. Contrairement aux technologies propriétaires, la plupart grande majorité des navigateurs du marché va les implémenter. Cela rendra les techniques utilisables par n'importe quel navigateur, ce qui était loin d'être le cas au début des années 2000. Nous détaillons dans la section qui suit un exemple de spécifications que traite le W3C, mais il y en a bien sûr beaucoup d'autres sur le site du Consortium.

### **2.5.1 CSS Media Queries**

Les *Cascading Style Sheets (CSS)*, ou feuilles de style en cascade constituent un des éléments incontournable dans le monde Internet. En découplant la partie présentation de celle du contenu des pages HTML, elles permettent de faciliter grandement la maintenance des applications Web en centralisant les instructions de styles qui étaient autrefois dupliquée et hard codées au sein de toutes les pages du site.

Les Media Queries, qui en sont encore en 2011 au stade de *Candidate Recommendation*, la troisième étape sur cinq du processus de standardisation du W3C, sont supportées par les versions récentes des navigateurs. Elles font partie de CSS3 et étendent le travail débuté par CSS2.

Le but des Media Queries est de déterminer de manière précise quelles sont les contraintes imposées par le média sur lequel une page web est consultée afin de fournir une feuille de style optimisée pour l’affichage. Pour ce faire, une media query est divisée en deux parties (4).

- **Un type de média** : déjà présent avec CSS2, cet attribut permet de déterminer quelle sera la catégorie de média utilisée et d’y attacher une feuille CSS particulière. Bien qu’étant un premier pas vers une adaptation efficace, on remarque vite que la précision est limitée. En effet, peut-on regrouper Smartphones et tablettes dans une seule catégorie *handheld* ? Les tailles d’écran sont relativement différentes et il serait judicieux de bénéficier d’autres mesures qui permettraient d’affiner le niveau de détails. Le tableau ci-dessous (5) reprend les différentes valeurs admises pour l’attribut `media` d’une balise `<link>` HTML.

Valeur de l’attribut <code>media</code>	Signification
<b>all</b>	Tout type de média
<b>aural (CSS 2.0) / speech (CSS 2.1)</b>	Synthèses vocales
<b>braille</b>	Plages braille
<b>embossed</b>	Imprimantes braille
<b>handheld</b>	Périphériques mobiles ou de petite taille
<b>print</b>	Impression
<b>projection</b>	Projecteurs (ou présentations avec slides)
<b>screen</b>	Écrans
<b>tty</b>	Terminal/police à pas fixe
<b>tv</b>	Téléviseur

Tableau 1 - Valeurs que peut prendre l’attribut `media` de la balise HTML `<link>`

L’exemple suivant montre comment se servir de cet attribut pour attacher des styles différents à un même contenu. Lorsque le type de média détecté est un écran normal, la feuille `screenStyle.css` est sélectionnée. Le même raisonnement peut être appliqué pour toute autre valeur de cet attribut.

```
<link rel="stylesheet" type="text/css" media="screen" href="screenStyle.css">
<link rel="stylesheet" type="text/css" media="handheld" href="handeldStyle.css">
```

- **0 ou plusieurs conditions** : introduites par CSS3, elles viennent véritablement apporter un degré de précision supplémentaire. Nous avons la possibilité de tester si des conditions très précises sont remplies. L'écran est-il en couleur ou noir et blanc ? A-t-il une taille inférieure à 480 pixels ? La page est-elle visionnée en mode portrait ou paysage ? L'écran est-il de type 4/3 ou 16/9 ? Telles sont les informations que l'on pourra obtenir et qui nous aiderons de manière significative à traiter plusieurs cas de figure. La liste (5) ci-dessous, plus complète, nous donne une idée des possibilités offertes.

Condition	Signification
<b>Color</b>	Support de la couleur (bits/pixel)
<b>color-index</b>	Périphérique utilisant une table de couleurs indexées
<b>aspect-ratio</b>	Ratio du périphérique de sortie (par exemple 16/9)
<b>device-aspect-ratio</b>	Ratio de la zone d'affichage
<b>device-height</b>	Dimension en hauteur du périphérique
<b>device-width</b>	Dimension en largeur du périphérique
<b>grid</b>	Périphérique bitmap ou grille (ex : lcd)
<b>height</b>	Dimension en hauteur de la zone d'affichage
<b>monochrome</b>	Périphérique monochrome ou niveaux de gris (bits/pixel)
<b>orientation</b>	Orientations du périphérique (portrait ou paysage)
<b>resolution</b>	Résolution du périphérique
<b>scan</b>	Type de balayage des téléviseurs (progressive ou interlace)
<b>width</b>	Dimension en largeur de la zone d'affichage

Tableau 2 - Liste des différents critères pouvant être utilisés dans des conditions

Dans la pratique, chaque media query consistera en une expression logique qui sera évaluée et retournera un résultat binaire VRAI ou FAUX. Plusieurs media queries peuvent exister au sein d'une même page, permettant d'adapter cette dernière à de multiples situations.

Le W3C propose de nombreux exemples sur la page de la spécification (4). Nous en avons repris deux pour illustrer ce que sont en pratique les media queries. Le premier exemple est une expression très simple qui ressemble à celle déjà présente en CSS2. Nous allons

appliquer la feuille de style « color.css » dans le cas où le média de destination est un écran standard et qui, de plus, supporte les couleurs.

```
<link rel="stylesheet" media="screen and (color)" href="color.css" />
```

Nous pouvons également imaginer un cas de figure dans lequel nous désirons appliquer une feuille de style particulière pour tout affichage sur écran ayant une largeur comprise entre 400 et 700 pixels, comme illustré ci-dessous.

```
<link rel="stylesheet"
      media="screen and (min-width: 400px) and (max-width: 700px)"
      href="example.css"
/>
```

Enfin, il est nécessaire de signaler que même si les media queries ne sont toujours pas une spécification officielle du W3C, des sites utilisent déjà ce système pour adapter dynamiquement l'apparence de leurs pages en fonction de l'environnement de l'utilisateur. Ainsi, l'image ci-dessous montre comment évolue la page du site <http://www.juslisen.com/> lorsque l'utilisateur modifie la largeur de la fenêtre du navigateur. Aucun rafraichissement de page n'est nécessaire, tout est question de sélection du fichier CSS approprié. On remarquera que lorsque la taille est de 320 pixels, le site est vraiment adapté pour une navigation mobile (pas d'image en arrière plan, pas de publicité).

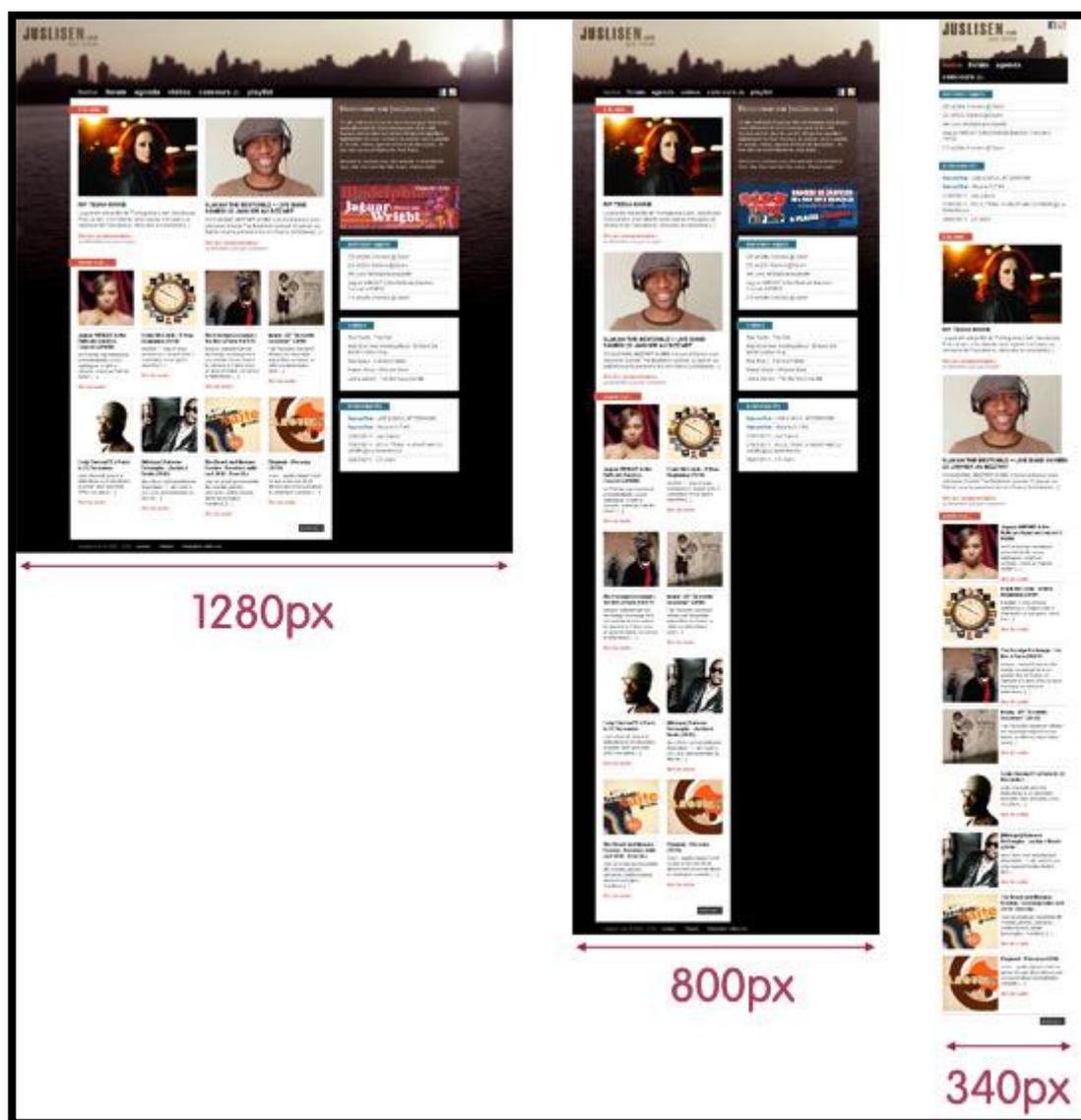


Figure 6 - Exemple d'adaptation d'une page Web via CSS Media Queries (image (5))

## 2.6 Présentation des techniques

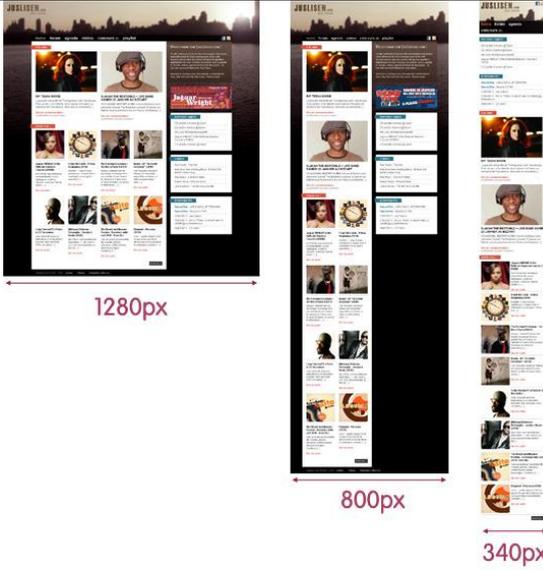
Nous présenterons les techniques d'adaptation en fonction de leur appartenance aux différentes catégories susmentionnées. Ces techniques sont toutes issues de l'article du W3C « Authoring Techniques for Device Independence » (3). Cet article se décompose en deux parties qui nous intéressent : d'une part il décrit des techniques d'adaptation précises, applicables aux sites Internet ; et d'autre part il contient une annexe dans laquelle sont rassemblées des techniques sous-jacentes plus basique et simplistes mais qui constituent véritablement les fondations de procédés plus évolués.

Pour décrire chaque technique, nous utiliserons un template générique facilitant la lecture et permettant à d'autres personnes travaillant sur le même sujet de s'y retrouver

rapidement. Chaque « fiche » sera rédigée en anglais afin de répondre à des critères de rédaction utilisés au sein du laboratoire de recherche LILab (Louvain Interaction Laboratory).

### 2.6.1 Style

Name	Context-Sensitive Style Selection
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#contextSensStyle">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#contextSensStyle</a>
Description	In order to adapt our webpage to a lot of different destination devices, we can define specific styles matching those devices display features. Styles are defined by CSS files and we will use CSS Media Queries to determine the device capabilities and provide the appropriate style sheet dynamically.
Rationale	We determine what the context is by using media queries, and then we provide the appropriate style sheet.
Example	We want to adapt a web page for a mobile phone. We create a specific style for it (e.g one column with less useless content like ads etc.) and link it via media queries.
Context	According to the context of use (size of the screen, color support, resolution)
Advantages	<ul style="list-style-type: none"> <li>- The display is dynamically adapted</li> <li>- Easy to use and understand</li> <li>- Can handle a lot of situations</li> </ul>
Disadvantages	The author must have a good knowledge of the different devices that can access his website and their features. There can be a lot of cases to deal with, resulting in a lot of different style sheets.
Sample	<pre>&lt;link rel="stylesheet"       media="screen and (max-width: 340px)       href="mobile.css" /&gt;</pre>

Pictures	
Observation	

Name	Scripted Style Selection
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#scriptedStyle">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#scriptedStyle</a>
Description	The style of elements delivered can be modified by scripts executed on the client side.
Rationale	The author can write some scripts that modify the style of some elements when some defined events occur.
Example	The overall content can be reduced by changing the style of some elements to invisible. For example, with JavaScript.
Context	<p>According to :</p> <ul style="list-style-type: none"> <li>- the context of use</li> <li>- the user preferences and actions</li> <li>- the destination client</li> </ul>
Advantages	The user has a decision power because styles are changed when some events occur (e.g. clicks, moving mouse's pointer on a specific area etc)
Disadvantages	This technique is not viable because it depends on the client's device and capabilities. The author has to know the specifications of different devices and create corresponding scripts. Furthermore some devices doesn't support some scripted stuffs and events.
Sample	<p>Javascript sample :</p> <pre>&lt;a href="#"   onclick="</pre>

	<code>document.getElementById(«elemToHide»).style.display="none";"</code> >Hide this part</a>
Pictures	
Observation	

Name	<b>Server-side Styling</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#serverSideStyling">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#serverSideStyling</a>
Description	The server (or an intermediate) integrates the styling information directly into the HTML pages.
Rationale	In place of having in one hand HTML content and in the other and CSS files, the CSS content is translated into HTML presentation tags and integrated into HTML page.
Example	This operation is very useful when the destination device doesn't support styles. We can determine this fact by using media queries in conjunction with UAProf or CC/PP
Context	According to the context of use.
Advantages	The content is available to a larger set of users and the design is preserved.
Disadvantages	<ul style="list-style-type: none"> <li>- The author must test and manage this adaptation process to be sure that the final result is correct</li> <li>- For some CSS properties, there is no corresponding HTML presentation possibilities</li> </ul>
Sample	<code>&lt;font size="3" color="red"&gt;My Text Here&lt;/font&gt;</code>
Pictures	
Observation	Even if the presentation HTML tags are deprecated, this solution is very useful in such a case.

Name	<b>Xforms Styling</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#xformsStyling">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#xformsStyling</a>
Description	Xforms introduces a presentation attribute « appearance » which can take 3 different defined values (user can create other values but there is no predefined behavior in this case so they have to be designed via CSS.). It can be applied to all the elements of a form.

Rationale	<p>There are three predefined values that have different effects depending on the form element on which they are applied. In general,</p> <ul style="list-style-type: none"> <li>- <u>Full</u> : The original presentation of the element</li> <li>- <u>Compact</u> : Like the full representation but with less details</li> <li>- <u>Minimal</u> : The element with the fewest detail degree</li> </ul>
Example	This feature can be used on small screens where the “minimal” value will be assigned to the appearance attribute
Context	According to the context of use, the preferences of the user
Advantages	This technique is flexible and can be used in a lot of different situations
Disadvantages	There are a lot of different implementations of Xforms (7). This technology is not natively implemented in all the browsers.
Sample	<p>With Xforms, there is a new form element called « select1 ». It allows the user to select a value among a given set.</p> <pre>&lt;select1 ref="color" appearance="full compact minimal"&gt;   &lt;label&gt;Favorite color ? :&lt;/label&gt;   &lt;item&gt;&lt;label&gt;Blue  &lt;/label&gt;&lt;value&gt;B&lt;/value&gt;&lt;/item&gt;   &lt;item&gt;&lt;label&gt;Red   &lt;/label&gt;&lt;value&gt;R&lt;/value&gt;&lt;/item&gt;   &lt;item&gt;&lt;label&gt;Green &lt;/label&gt;&lt;value&gt;G&lt;/value&gt;&lt;/item&gt;   &lt;item&gt;&lt;label&gt;Yellow&lt;/label&gt;&lt;value&gt;Y&lt;/value&gt;&lt;/item&gt; &lt;/select1&gt;</pre>
Pictures	<div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>Favorite color ?</p> <p><input checked="" type="radio"/> Blue  <input type="radio"/> Red  <input type="radio"/> Green  <input type="radio"/> Yellow</p> <p><b>FULL</b></p> </div> <div style="text-align: center;"> <p>Favorite color ?</p>  <p><b>COMPACT</b></p> </div> <div style="text-align: center;"> <p>Favorite color ?</p>  <p><b>MINIMAL</b></p> </div> </div>
Observation	

### 2.6.2 Layout

Nous pouvons utiliser une combinaison de HTML et CSS pour déterminer le layout. En utilisant les média queries, il est possible d'associer un style différent aux éléments `<div>`, qui définissent la position et la taille des blocs d'informations, en fonction du contexte. On pourra donc les rendre plus ou moins grands, invisibles ou non etc. La technique

des CSS média queries ayant déjà été présentée dans la section « style », nous ne reviendrons pas dessus dans cette partie.

### 2.6.3 Contenu

Les techniques concernant le contenu, présentées dans le document du W3C, se concentrent fortement sur une approche orientée sélection. Le principe de ce genre de technique est de déterminer quelles sont les contraintes imposées par l'environnement dans lequel est visionnée une page Web et de sélectionner en conséquence les éléments adaptés pour les présenter correctement à l'utilisateur.

Name	<b>Variant Selection with SMIL &lt;switch&gt; element</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#contentSMIL">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#contentSMIL</a>
Description	There exists a <switch> element in SMIL that offers to the author the possibility to select a resource among a given set of alternatives.
Rationale	The <switch> element has the same working principle than its corresponding instruction in classical programming languages. Each condition is evaluated and if it is true, the corresponding statement is executed. A default case is executed when no condition were satisfied.
Example	This element could be use for adaptation purpose. We can test for example what is the screen size of the destination device and select appropriate content for a quality presentation.
Context	According to the context of use, the preference of the user.
Advantages	<ul style="list-style-type: none"> <li>- Very simple and comprehensive syntax</li> <li>- Could be used for automated adaptation process</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>- SMIL is not yet natively implemented in all the browsers of the market (summer 2011).</li> <li>- If there is a lot of elements that have to be adapted, there will be a lot of &lt;switch&gt; elements</li> </ul>
Sample	<pre>&lt;switch&gt;   &lt;img src="img_1024.jpg" systemScreenSize="1024X1280"/&gt;   &lt;img src="img_480.jpg" systemScreenSize="480X640"/&gt;   &lt;img src="img_240.jpg" systemScreenSize="240X320"/&gt;   &lt;img src="img_default.jpg"/&gt; &lt;/switch&gt;</pre>
Pictures	

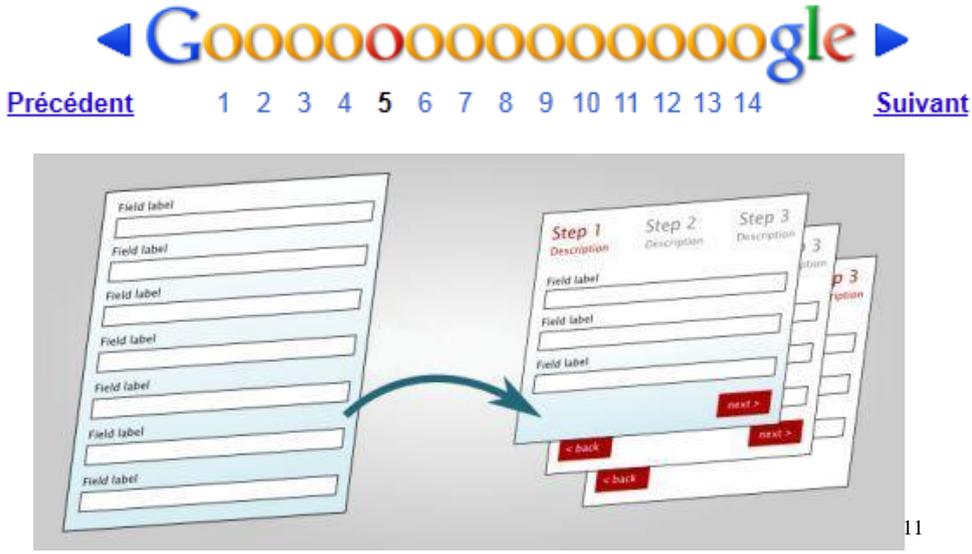
Observation	Although this solution is working, content adaptation is not the final purpose of SMIL.
-------------	---

Les CSS media queries peuvent également être utilisées pour ce type d'adaptation.

#### 2.6.4 Structure

Nous nous pencherons sur les techniques de décomposition qui permettent de scinder un contenu global en plusieurs fragments. Ce type de technique altère la structure globale du document puisque les parties décomposées ont chacune leur propre structure. Il faut également se poser la question du lieu de la scission. Où scinder ? Quand scinder ? Comment relier entre eux les différents morceaux résultant ?

Name	<b>Pagination and page numbers</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structurePagination">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structurePagination</a>
Description	The content is divided into a collection or ordered smallest pages. The division is made according to a mathematical expression (e.g. 5 news per page, 40 lines per page, etc.).
Rationale	Several cases are possible. We can detect the capabilities of the destination device and calculate a good ratio content/page to provide or the user can define it according to his preferences. We also can define a static value for pagination.
Example	We can use pagination to adapt content to small screen devices. Pagination can take different forms : <ul style="list-style-type: none"> <li>- <u>Paper pages</u>: The content is automatically divided according to the size of pages to be printed.</li> <li>- <u>Tabbed pages</u> : A kind of menu composed of tabs allows the user to select which page to view</li> <li>- <u>Wizards</u>: A form is divided into several parts that can be viewed as steps. The user has to complete a step before going to the next one.</li> </ul>
Context	According to the context of use, the preferences of the user.
Advantages	<ul style="list-style-type: none"> <li>- Simple and powerful concept.</li> <li>- Can be easily put in place.</li> <li>- Flexible</li> </ul>

Disadvantages	<ul style="list-style-type: none"> <li>- The criteria for pagination has to be well defined</li> <li>- There can be a lot of sub-parts in the case of really small screen</li> </ul>
Sample	jQuery FormWizard plugin <sup>10</sup>
Pictures	 <p>The image shows a pagination interface for Google search results. At the top, there is the Google logo with a left arrow and a right arrow. Below the logo, there are links for 'Précédent' and 'Suivant', and a list of numbers from 1 to 14. Below this, there is a diagram of a multi-step form wizard. The diagram shows a stack of three steps, each with a 'Field label' input field. The first step is labeled 'Step 1' and has a 'next &gt;' button. The second step is labeled 'Step 2' and has a 'next &gt;' button. The third step is labeled 'Step 3' and has a 'next &gt;' button. There are also 'back &lt;' buttons for each step. The number '11' is visible in the bottom right corner of the diagram.</p>
Observation	

L'auteur d'un document peut influencer le processus de pagination en utilisant des procédés couramment utilisés dans les traitements de texte (marquage de fin de paragraphe, délimitation de sections bien définies, etc.)

Name	<b>Page Breaks</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structureADDecomposition">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structureADDecomposition</a>
Description	The author divides the content with special marks.
Rationale	The specific marks can be processed automatically to divide the content into smallest pieces.
Example	A system can take these marks as separator and cut the content for mobile adaptation.
Context	According to the context of use, the marks, and the preferences of the user.
Advantages	Marks can be processed automatically
Disadvantages	If the case of big content, the author can spend a lot of time to put the marks

<sup>10</sup> <http://plugins.jquery.com/project/formwizard>

<sup>11</sup> Image adaptée de : <http://www.jankoatwarpspeed.com/post/2009/09/28/webform-wizard-jquery.aspx>

Sample	The <hr> html tags
Pictures	
Observation	

Name	<b>Widow/Orphan control</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structureADDecomposition">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structureADDecomposition</a>
Description	We want to avoid widows and orphans
Rationale	<ul style="list-style-type: none"> <li>- Widow : 1 line on page A and N-1 lines on page B</li> <li>- Orphan : N-1 lines on page A and 1 line on page B</li> </ul>
Example	For small screen destination devices, it could be interesting to view entire parts of the content rather than a mix of different pieces.
Context	According to the context of use, the preferences of the user.
Advantages	The display is optimized to see full parts of a paragraph
Disadvantages	There can be pages that contain less data than they could contain
Sample	
Pictures	
Observation	

Name	<b>Sectioning</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structureADDecomposition">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structureADDecomposition</a>
Description	Content can be divided according to sections and subsections marks

Rationale	An HTML document can contain several tags defining sections, subsections and so on. We can divide this document by using these tags as reference.
Example	In a mobile environment with small screen, decomposing a document according to these tags could represent a first step in the adaptation process.
Context	According to the context of use, the preferences of the user.
Advantages	Most of the authors use these native tags to create documents. This represents a natural structure that can be used for decomposition. Flexible solution
Disadvantages	This technique has to be combined with other ones because sections can be huge
Sample	The heading HTML tags (<h1>, <h2>, <h3>...)
Pictures	<p>The diagram shows a large document on the left with three sections separated by H1 tags. Arrows point from these sections to three smaller, stacked document fragments on the right, labeled 1, 2, and 3, representing the decomposed content units.</p>
Observation	

Name	<b>Regions</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structureADDecomposition">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structureADDecomposition</a>
Description	A web page can be viewed as a set of regions which contain some content. Decomposition can take these regions as content units.
Rationale	Given an HTML document, we decompose it according to its regions to obtain smallest parts that can be delivered in a given context.
Example	Content decomposition for small screen devices.
Context	According to the context of use, the preferences of the user.
Advantages	Most of the authors use these native tags to create documents. This represents a natural structure that can be used for decomposition. Flexible solution
Disadvantages	This technique has to be combined with other ones because regions can be

	huge
Sample	The <div> and <span> HTML tags
Pictures	
Observation	

### 2.6.5 Navigation

Name	Access Keys
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structNavAuthored">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#structNavAuthored</a>
Description	Access Keys are a one-touch access to a resource. The idea is to associate a defined keyboard key with a link or region of a website to improve the navigation in particular contexts.
Rationale	Several keyboard keys (e.g. numerical keys) are associated to an action/region on the web page. Clicking on such keys is equivalent to clicking on the target link.
Example	This can be very useful in presence of devices without touch screen or for people with reduced movement capabilities.
Context	According to the context of use or the preferences of the user.
Advantages	This technique is very useful when browsing the web with mobile phone with just a limited keyboard as interface.
Disadvantages	This can be difficult to point the relevant parts of a website. The consequence is that some important parts (login link) are not selected while irrelevant content is mapped. A better solution should be based on user experience to improve the mapping. If the content is big, there can be a lot of access keys needed.
Sample	

Pictures	<p>An example of the UCL website adapted by the Google Mobilizer tool.</p> 
Observation	

### 2.6.6 Interaction

Name	<b>Form Resizing</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#csaFormResize">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#csaFormResize</a>
Description	Forms can be complicated, contain a lot of fields and in some cases cannot be displayed entirely on the screen (Smartphones etc). Several operations can be used to solve this problem.
Rationale	<p>Given a form, different alternatives (that can be combined) can be used to reduce its size :</p> <ul style="list-style-type: none"> <li>- <u>Reducing the form's controls visualization</u> : The font size of form controls can be reduced to reduce the global space taken by the form</li> <li>- <u>Repositioning form controls</u>: By reducing space between form controls etc. We have to be careful to not create overlapping.</li> <li>- <u>Transforming form controls</u> : A control can be transformed into another one that takes less space (see picture and Xforms Styling)</li> <li>- <u>Splitting the form</u>: In the last resort, we can split the form. This operation is complex because we have to group several form controls together, respect constraints and define an order to navigate through the form.</li> </ul>
Example	In a reduced display screen environment, we can reduce the size of the form in order to be usable by the end-user.
Context	According to the context of use, the preferences of the user, the size of the screen
Advantages	This technique is mandatory for small screen devices.

Disadvantages	The process can be difficult to implement, raise some errors or bring confusion to the user.
Sample	Xforms, jQuery plugins, ...
Pictures	<p>Pictures from the W3C illustrating the transformation of form controls</p>
Observation	The “Pagination and page numbers” technique could be used for form splitting while the transformation of form controls can be achieved by the Xforms styling technique (see above).

Name	Interaction Session
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#interactionSession">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#interactionSession</a>
Description	Sessions enable the client to have a persistent link with the distant server. Without this mechanism, it wouldn't be possible to login to a website.
Rationale	This mechanism needs the cookies to be activated to work. If the destination device doesn't support cookies, other solution must be employed. A common solution is to put the session key in the URL.
Example	Several devices don't support cookies, or the user has disabled this functionality.
Context	According to the device capabilities or the preferences of the user
Advantages	The client is able to use sessions
Disadvantages	The transmission of the session key in the URL is less secured
Sample	<a href="http://www.mysite.be/page.php?session_id=rj3ids98pas2rouf3jc89mq1t0">http://www.mysite.be/page.php?session_id=rj3ids98pas2rouf3jc89mq1t0</a>
Pictures	
Observation	

### 2.6.7 Techniques d'adaptation sous-jacentes

Les techniques que nous avons présentées jusqu'à présent ne sont pas toutes primaires dans le sens où elles peuvent dépendre et se reposer sur une ou plusieurs autres techniques qui

traitent d'un problème plus spécifique. Nous présentons ici ces solutions plus basiques qui permettent d'adapter du contenu en fonction du contexte. Toutefois nous ne les présenterons pas toutes pour des raisons de clarté. Cela sera le cas notamment pour les techniques que nous avons déjà présentées dans les sections qui précèdent (ex : sélection de style via CSS media queries).

### 2.6.7.1 *Select/Remove*

Le principe est simple mais efficace : en fonction du contexte, certaines parties du contenu sont sélectionnées pour être présentées tandis que d'autres sont omises.

Name	<b>Relevance of Content</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#relevanceOfContent">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#relevanceOfContent</a>
Description	Xforms introduces a new attribute "relevant" that can be inserted into XML tags to provide some information about relevant content in a given context.
Rationale	This attribute takes as value a Boolean expression that can lead to different ways : <ul style="list-style-type: none"> <li>- Expression = true → content is selected for presentation</li> <li>- Expression = false → content isn't included in the user display</li> </ul>
Example	We can use expression to test the capabilities of the destination device (screen size, color support etc) and select what is the best content to provide.
Context	According to the context of use, the preferences of the user.
Advantages	Because we can put the attribute into the markups, we can precisely define what parts of the content will be selected/removed in any cases.
Disadvantages	It's possible to have some duplication of code because of the fact that the "relevant" attribute is put among the tags.
Sample	An example from the W3C document (3) <pre>&lt;xhtml&gt;   &lt;img src="a" DI:relevant="Context('colorDepth')   &amp;lt; 257" /&gt; &lt;/xhtml&gt;</pre>
Pictures	
Observation	

### 2.6.7.2 Navigation generation

La navigation est un des enjeux majeurs des sites Internet. En effet, peu importe le type de média utilisé, la navigation doit véritablement permettre à l'utilisateur d'avoir un accès rapide aux ressources qu'il cherche. Une navigation pauvre peut entraîner une certaine frustration chez l'utilisateur qui soit ne trouvera pas ce qu'il cherche, soit devra réaliser un certain nombre d'étapes avant d'y accéder. Si cela est vrai dans des cas d'utilisation normaux (consultation sur un écran de bureau), ça l'est encore plus dans un environnement embarqué avec des écrans de taille réduite et une connexion potentiellement plus lente.

Nous détaillons des techniques permettant de créer une navigation en fonction du contenu ou d'informations spécifiées par l'auteur.

Name	<b>Table of Content</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#navGenTOC">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#navGenTOC</a>
Description	A table of contents is automatically generated and provides fast access to different parts of a given page. The table is generally hierarchical and presents the sections in the order they appear in the document.
Rationale	The table can be generated from the <h1>, <h2> ... HTML tags
Example	This can be used in several cases : <ul style="list-style-type: none"> <li>- The page contains a lot of information and a table of contents will help the user to navigate through and understand the structure of the document</li> <li>- If we access the page with small screen device, the table of contents will avoid us to load useless data. We will directly go to the part of the content that is interesting for us.</li> </ul>
Context	Small screen, low connection, preferences of the user
Advantages	<ul style="list-style-type: none"> <li>- Very simple to generate</li> <li>- If the content is updated, the table is updated</li> <li>- Give a good vision of the overall content</li> <li>- Enable the user to gain some time (searching information he wants and connection time)</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>- Cannot be applied in all cases</li> </ul>
Sample	
Pictures	A Wikipedia table of contents

	<b>Contents [hide]</b>
	<ul style="list-style-type: none"> <li>1 History <ul style="list-style-type: none"> <li>1.1 Major achievements</li> </ul> </li> <li>2 Philosophy</li> <li>3 Areas of computer science <ul style="list-style-type: none"> <li>3.1 Theoretical computer science <ul style="list-style-type: none"> <li>3.1.1 Theory of computation</li> <li>3.1.2 Information and coding theory</li> <li>3.1.3 Algorithms and data structures</li> <li>3.1.4 Programming language theory</li> <li>3.1.5 Formal methods</li> <li>3.1.6 Concurrent, parallel and distributed systems</li> <li>3.1.7 Databases and information retrieval</li> </ul> </li> <li>3.2 Applied computer science <ul style="list-style-type: none"> <li>3.2.1 Artificial intelligence</li> <li>3.2.2 Computer architecture and engineering</li> <li>3.2.3 Computer graphics and visualization</li> <li>3.2.4 Computer security and cryptography</li> <li>3.2.5 Computational science</li> <li>3.2.6 Information science</li> <li>3.2.7 Software engineering</li> </ul> </li> </ul> </li> </ul>
Observation	

Name	<b>Nearest Neighbor and Home</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#navGenNearest">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#navGenNearest</a>
Description	The idea is to provide to the user a suitable homepage and a menu containing links that point to relevant content according to the information he's currently viewing.
Rationale	The menu can be generated from the hierarchical site structure or defined by the webmaster. In the first case, the determination of the menu's elements (i.e. nearest neighbors) will return the parent node and its children nodes.
Example	<ul style="list-style-type: none"> <li>- This can be very useful in big sites where navigation represents a challenge</li> <li>- These relevant links can accelerate the navigation on small screen devices</li> </ul>
Context	According to the site structure, the preferences of the user
Advantages	<ul style="list-style-type: none"> <li>- The links are relevant with regard to the context and are therefore</li> </ul>

	<p>helpful to the user</p> <ul style="list-style-type: none"> <li>- The menu can be automatically generated</li> </ul>
Disadvantages	In a small screen context, the menu must be kept small
Sample	
Pictures	
Observation	

Name	<b>Full Site Representation</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#navGenFullSite">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#navGenFullSite</a>
Description	This is a sitemap representation. We represent the entire structure of the website independently of the context of use (current page, user preferences)
Rationale	The global structure of the website (e.g. set of all the headings) is displayed and provides direct access to different sections.
Example	The sitemap can represent a starting point from which users can navigate
Context	The preferences of the user,
Advantages	<ul style="list-style-type: none"> <li>- Fast access to all the parts of the website</li> <li>- Global view of the website</li> <li>- The map can be reduced by collapsing some tree sections</li> </ul>
Disadvantages	The full site representation is usually big and is not suitable for small screens display.
Sample	
Pictures	An example from <a href="http://siteduzero.com">siteduzero.com</a> (only a part of the tutorials section is

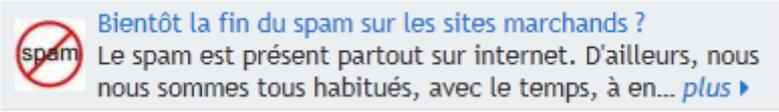
	<h2 style="text-align: center;">Tutoriels</h2> <ul style="list-style-type: none"> <li>○ Accueil des tutoriels</li> <li>○ Tutoriels       <ul style="list-style-type: none"> <li>○ Site Web           <ul style="list-style-type: none"> <li>○ XHTML / CSS</li> <li>○ PHP               <ul style="list-style-type: none"> <li>○ Base de données (PHP)</li> <li>○ Sécurité (PHP)</li> <li>○ Points particuliers (PHP)</li> <li>○ Systèmes complets (PHP)</li> <li>○ Général (PHP)</li> </ul> </li> <li>○ Divers</li> <li>○ Forums</li> <li>○ Hébergement</li> <li>○ Javascript</li> <li>○ Serveur</li> <li>○ IDE</li> </ul> </li> <li>○ Programmation           <ul style="list-style-type: none"> <li>○ AS / Flash / Flex</li> <li>○ Algorithmique</li> <li>○ Basic et dérivés</li> <li>○ Bibliothèques               <ul style="list-style-type: none"> <li>○ API Windows</li> </ul> </li> </ul> </li> </ul> </li> </ul> <p style="text-align: center;">shown)</p>
Observation	

Name	<b>Next and Previous Links</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#navGenNextPrevious">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#navGenNextPrevious</a>
Description	« The purpose of Next and Previous links it to facilitate linear navigation to adjacent content » (3)
Rationale	<p>The adjacent content can be interpreted in different ways. According to the sense we give to this concept, we can obtain 2 versions of next and previous links.</p> <ul style="list-style-type: none"> <li>- <u>Relative to current path</u>: this is the implementation made in the browsers for the next and previous buttons. When browsing a site, the navigation path created can be used to go back or forward.</li> <li>- <u>Relative to site structure</u>: The structure can be defined by the author or created from a document (powerpoint, etc.). Next and previous links will only be focused on the structure and not on the navigation path.</li> </ul>
Example	<ul style="list-style-type: none"> <li>- There are few links so it can be used in small screens environment</li> <li>- It's pretty useful for a presentation with linear progression</li> </ul>

Context	According to the context of use, the preferences of the user
Advantages	<ul style="list-style-type: none"> <li>- Easy to use and intuitive</li> <li>- Easy to implement</li> </ul>
Disadvantages	The user can have the feeling to be locked in a linear path. By providing some other useful links (like the “up” link on the picture) we can decrease the degree of this disadvantage
Sample	
Pictures	<p>In this example relative to site structure :</p> <ul style="list-style-type: none"> <li>- The current page is “Aire d’une region de <math>R^2</math>”</li> <li>- Next and previous pages are surrounding the current page</li> <li>- All of those pages are contained in the section “Intégrale double”.</li> </ul> <p>Clicking on this link bring the user to the plan of the section.</p> <p><b>Next:</b> <a href="#">Cas général de l'intégrale</a> <b>Up:</b> <a href="#">Intégrale double</a> <b>Previous:</b> <a href="#">La somme des volumes</a></p> <p><b>Aire d'une région de <math>R^2</math></b></p>
Observation	The “next and back links” technique is a mix the current we are describing. In this technique, the next page is defined in advance and the back page is the last viewed page.

### 2.6.7.3 Substitution

Name	<b>Substitution</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#sec13AdaptationViaSubstitution">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#sec13AdaptationViaSubstitution</a>
Description	Several parts of the content are substituted by smaller content
Rationale	The substituted content is replaced by a link pointing to the full version.
Example	<ul style="list-style-type: none"> <li>- Replacing some special characters that couldn’t be supported by all devices (€ → EUR)</li> <li>- Reduce the text size for small screen devices</li> </ul>
Context	According to the context of use, the preferences of the user
Advantages	<ul style="list-style-type: none"> <li>- The size of the content is drastically reduced</li> <li>- A good practice for mobile adaptation</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>- This process can be automated but we have to be careful to the</li> </ul>

	semantic of the content that could be modified. (“They eat Belgian food when they...” → “They eat Belgian...”)
Sample	
Pictures	
Observation	

#### 2.6.7.4 Server-side adaptation

Name	<b>Transformation</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#ssaTransformation">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#ssaTransformation</a>
Description	The document and its structure are transformed.
Rationale	Transformation can take several forms : <ul style="list-style-type: none"> <li>- Transforming XML to SVG, HTML to WML<sup>12</sup>, HTML filtering</li> <li>- Transformation operations on media (videos etc.)</li> <li>- Advanced transformation (relative to the content itself, for example summarization of content) is also possible but requires some semantic annotations from the author (the idea of semantic web)</li> </ul>
Example	Transformation is often necessary when the content is not suitable for mobile devices
Context	According to the context of use
Advantages	- The result is fast and the process automated
Disadvantages	- Designing transformation algorithm requires some programming skills - Transformation is an automated process and can bring some errors or inconsistencies, because it considers general cases.
Sample	This task can be achieved with XSLT
Pictures	
Observation	

Name	<b>Media Adaptation</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#ssaMediaAdatation">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#ssaMediaAdatation</a>

<sup>12</sup> Wireless Markup Language

Description	All the media (videos, images, audio) in a web page can be transformed and adapted according to the delivery context
Rationale	There are a lot of possible operations (resizing, re-encoding, color reduction, compression etc.)
Example	These operations are suitable for mobile access (resizing for small screens, re-encoding to a more common format, compression for low connections)
Context	According to the context of use, for low connection speed, small screen devices
Advantages	<ul style="list-style-type: none"> <li>- The media files are the biggest part of the content, in terms of memory space needed. Reduce their size and weight is essential for the mobile world</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>- This is hard to design and implement</li> <li>- Some information can be lost</li> </ul>
Sample	
Pictures	
Observation	This technique describes a general principle. There is a lot of techniques involved in “media adaptation”

Name	<b>Decomposition</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#adaptViaDecomposition">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#adaptViaDecomposition</a>
Description	The web page is decomposed into smaller parts
Rationale	<p>There are two forms of decomposition: into delivery units (DU) and presentation units (PU). These terms are defined in the W3C DI principles document (8)</p> <ul style="list-style-type: none"> <li>- <u>Decomposition into DU's</u>: A delivery unit is a “set of material transferred between two cooperating web programs as the response to a single HTTP request”. This is typically the document returned when a web page is requested. Dividing content into small DU's can be useful when the destination device has limited memory and storage capabilities (e.g. the deck size in the WML language, which can be limited to 1200 bytes).</li> <li>- <u>Decomposition into PU's</u> : The resource can be divided into several smaller Presentation Units. This process aims at presenting a content in an</li> </ul>

	appropriate manner for the user, according to the destination device
Example	<ul style="list-style-type: none"> <li>- By-pass the deck size limitation of WML</li> <li>- Avoid horizontal scrolling</li> </ul>
Context	According to the context of use (memory space on the client, screen size, etc.)
Advantages	Decomposed content is more suitable for mobile use Decomposition doesn't involve lose of information
Disadvantages	Content can be divided at wrong places
Sample	Delimitation of regions with <div> tags can help to determine the place to
Pictures	
Observation	

### 2.6.7.5 Client-Side adaptation

Name	<b>Image resizing</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#csaImage">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#csaImage</a>
Description	Images are resized to be viewed in a limited area.
Rationale	Image resizing can be automated but in several cases (e.g map resizing) some subtle mechanisms have to be used if we want the image to be still consistent
Example	Creation of thumbnails, adaptation to small screens and low bandwidth connection
Context	According to the context of use, the preferences of the user.
Advantages	Useful for small screen devices and poor connection network
Disadvantages	<ul style="list-style-type: none"> <li>- Information can be lost (too small to be viewed correctly)</li> <li>- Several cases are difficult to manage</li> </ul>
Sample	
Pictures	

Observation	
-------------	--

Name	<b>Font substitution</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#csaFont">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#csaFont</a>
Description	Different fonts can be used in a single document. But some mobile devices don't support all the fonts used on the Internet. Font substitution could solve this problem by replacing all the fonts in the document by a single standard font.
Rationale	Given a set of fonts, we substitute all of them by one or more standard fonts (times, courier, ...)
Example	Several fonts are hard to read or cannot be correctly displayed on all destination devices. In such cases we can replace these fonts with a standard one. It also could be useful for persons with reduced vision capabilities
Context	According to the context of use, the preferences of the user, the device capabilities
Advantages	A larger public can view the content
Disadvantages	Some styling information are lost
Sample	
Pictures	<p>The diagram illustrates the process of font substitution. It consists of two rows of boxes. Each row has an arrow pointing from a box on the left to a box on the right. In the top row, the left box contains the text 'Sample Sample Sample' in a serif font, and the right box contains the same text 'Sample Sample Sample' in a standard sans-serif font. In the bottom row, the left box contains the text 'Sample Sample Sample' in a cursive font, and the right box contains the same text 'Sample Sample Sample' in a standard sans-serif font. This demonstrates how different fonts are replaced by a single standard font for better readability on devices that may not support all fonts.</p>
Observation	

Name	<b>Dedicated rendering</b>
References	<a href="http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#csaRenderer">http://www.w3.org/TR/2004/NOTE-di-atdi-20040218/#csaRenderer</a>

Description	Several elements in the delivered content are specific. They need specialized programs to be read correctly.
Rationale	Small programs (like browser's plugins) could be installed on the client side to read this specialized content.
Example	Formats such PDF, Flash and so on require specific readers that are not always present on the destination devices.
Context	According to the destination device capabilities
Advantages	Improve the possibilities of the destination context
Disadvantages	These programs aren't always installed or available for installation on destination devices
Sample	
Pictures	
Observation	A server-side adaptation could transform the specialized content into a corresponding alternative format

### 2.6.8 Tableau récapitulatif des techniques théoriques

Catégorie	Techniques associées
Style	<ul style="list-style-type: none"> <li>- Context-Sensitive Style Selection</li> <li>- Scripted Style Selection</li> <li>- Server-side Styling</li> <li>- Xforms Styling</li> <li>- Font substitution</li> <li>- Form resizing</li> </ul>
Contenu	<ul style="list-style-type: none"> <li>- Variant Selection with SMIL</li> <li>- Relevance of Content</li> <li>- Substitution</li> <li>- Media Adaptation</li> <li>- Image resizing</li> <li>- Dedicated rendering</li> </ul>
Structure	<ul style="list-style-type: none"> <li>- Pagination and page numbers</li> <li>- Page breaks</li> <li>- Widow/Orphan Control</li> <li>- Sectioning</li> <li>- Regions</li> <li>- Transformation</li> <li>- Decomposition</li> <li>- Form resizing</li> </ul>
Navigation	<ul style="list-style-type: none"> <li>- Access Keys</li> <li>- Table of Content</li> <li>- Nearest Neighbor and Home</li> <li>- Full site Representation</li> <li>- Next and Previous Links</li> <li>- Form resizing</li> </ul>
Interaction	<ul style="list-style-type: none"> <li>- Form resizing</li> <li>- Interaction session</li> </ul>
Layout	<ul style="list-style-type: none"> <li>- CSS media queries (context-sensitive style selection)</li> <li>- SMIL &lt;layout&gt;</li> </ul>

### 3 Proposition de solution

Nous venons de passer en revue un ensemble de techniques permettant d'adapter différents aspects d'une page web pour que celle-ci s'adapte à un contexte donné. Ces techniques peuvent, pour la plupart, être combinées entre elles pour obtenir des solutions plus ou moins satisfaisantes.

Cependant, au vu des outils déjà existants sur le marché que nous avons eu l'occasion de parcourir au chapitre 2, il semblerait que des solutions de hautes qualité comme Skweezer ou encore « Opera mini » manipulent déjà toutes ces techniques de manière optimale. Nous n'avons pas la prétention de proposer une meilleure solution sur le plan technique que celles proposées par ces outils, qui sont développés depuis plusieurs années par des équipes de professionnels.

Au lieu de cela, nous pouvons nous pencher sur une autre facette de ce que peut représenter l'adaptation. Les solutions existantes reposent sur des processus automatisés qui prennent en entrée un ensemble de contraintes et un ensemble de données, et retourne un contenu adapté. Notre proposition consiste à changer les « règles du jeu ».

Nous avons fréquemment remarqué lors de notre phase de test des différents outils que le problème majeur était la quantité d'informations retournées. L'affichage se faisant en mode « une colonne », il est parfois nécessaire de scroller et de chercher parmi tout le texte condensé pour retrouver le lien ou l'information que l'on recherche. De plus, tout ce contenu représente un certain coût en termes de temps de chargement etc.

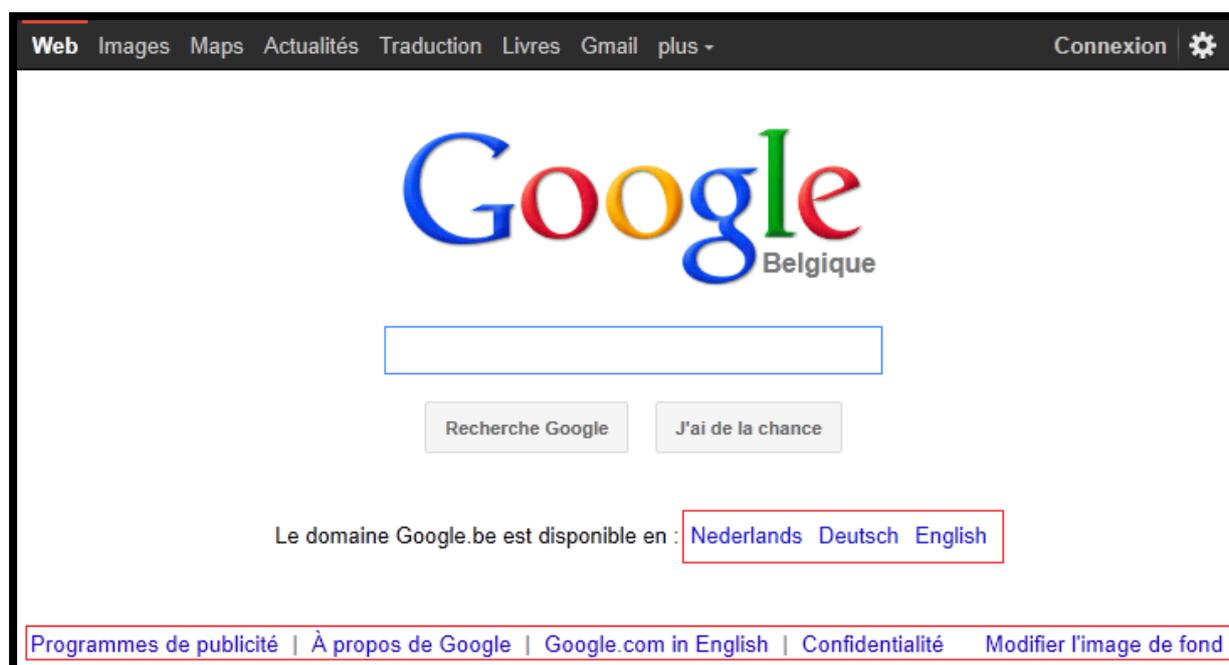
Notre solution consiste à introduire la notion de pertinence dans le processus d'adaptation, pour fournir à l'utilisateur un contenu épuré mais adapté à ses besoins.

#### 3.1.1 La notion de pertinence : méthode de détermination

La pertinence est une notion qui n'est pas nouvelle sur Internet. Google par exemple, classe ses résultats de recherche du plus pertinent au moins pertinent, selon une formule gardée secrète qui serait basée sur plus de 200 facteurs différents. Un de ces facteurs est le

nombre de liens pointant vers une page web : ces liens ont par définition été créés par des humains qui trouvaient que le contenu de la page était intéressant.

Nous pouvons partir de ce concept et nous en servir comme principe de base pour définir la pertinence : agréger les comportements individuels des utilisateurs en tendances globales de navigation. Illustrons cela par un exemple simple : l'utilisation du moteur de recherche Google. Chaque jour, des millions de personnes utilisent ce service et se rendent sur la page d'accueil du moteur de recherche.



Parmi ces utilisateurs, nous pouvons supposer sans trop de risque qu'il y en a peu qui vont cliquer sur les liens encadrés en rouge, par rapport au nombre qui lancent une recherche. Nous pourrions, à titre indicatif, récolter des statistiques et découvrir que 93% des utilisateurs lancent une recherche, 4% vont se connecter via l'onglet « connexion », 2 % se dirigeront sur google maps, et les 1 % restant effectueront une autre action.

Si nous pouvions disposer de tels chiffres, nous pourrions les intégrer à notre processus d'adaptation pour proposer une version pour mobile dans laquelle on ne retrouverait par exemple pas tous les liens encadrés en rouge. Comment dès lors, pouvons-nous obtenir ces informations ?

Deux sources de données peuvent nous aider à déterminer quels sont les éléments les plus pertinents d'un document Web :

- Les utilisateurs de notre solution : qui, de manière anonyme, nous laissent une trace de leur navigation sur le web à partir de laquelle nous pouvons établir des statistiques pour chaque site visité.
- Les webmasters : qui peuvent nous aider en nous envoyant des rapports détaillés des statistiques d'utilisation de leur site (qui peuvent être facilement déterminées à l'aide d'outils gratuits). Néanmoins nous pouvons nous attendre à ce que tous ne collaborent pas.

### 3.1.2 Principe général

Nous parlions de changer les règles du jeu mais il n'en n'est pas réellement question. Les travaux réalisés sur l'adaptation sont toujours valables et nécessaires dans notre approche. Nous ne faisons qu'ajouter un facteur supplémentaire dans le processus, en pondérant les éléments d'un document.

Notre proposition se repose sur un modèle de proxy, ce qui nous permettra d'avoir un contrôle total sur les opérations de transformation du contenu. De plus nous aurons la possibilité de récolter des données de navigation des utilisateurs dans le but de construire des statistiques et d'en tirer certaines tendances. Pour aller plus loin encore, nous pourrions offrir aux utilisateurs qui le souhaitent une expérience personnalisée en leur proposant un contenu adapté en fonction de leurs habitudes de navigation.

La possibilité de ne présenter que le contenu pertinent, c'est-à-dire populaire, est très intéressante mais il est impératif de ne pas négliger les comportements plus marginaux. Cette navigation améliorée doit être un « plus » mais ne doit pas être intrusive et définitive. Dans la pratique, l'utilisateur final doit toujours avoir le choix entre un contenu adapté de manière standard et automatique et un contenu adapté en fonction des tendances (ou encore un contenu adapté en fonction de ses habitudes).

### 3.1.3 Fonctionnement global

D'un point de vue technique, le processus de décision consistant à sélectionner une version donnée d'une page adaptée peut être représenté par le diagramme ci-dessous. Tout est effectué au niveau du proxy.

1. Lorsqu'un utilisateur envoie une requête à notre serveur, nous allons contrôler s'il a activé l'option « Trends ». Si tel est le cas, cela signifie qu'il désire obtenir une page dessinée selon les tendances de navigation. Cette option peut être activée par défaut. S'il ne l'a pas activée, nous lui retournerons la page adaptée de manière classique.
  
2. Le mode « trends » étant activé, nous vérifions si la page désirée est présente dans notre base de données et a déjà été visitées auparavant. Si ce n'est pas le cas, nous n'avons d'autres choix que de proposer une version adaptée standard.
  
3. Si la page est renseignée, nous devons vérifier si une double condition est remplie avant de déterminer quelle version proposée à l'utilisateur.
  - a. L'utilisateur désire-t-il voir une page adaptée selon ses préférences ? Si tel est le cas, a-t-il déjà visité suffisamment la page pour que l'on puisse en définir ses habitudes ? Si ces conditions sont réunies, nous lui fournissons une version personnalisée
  
  - b. L'utilisateur peut également choisir de garder un certain anonymat. Dans ce cas nous lui offrons la version de la page adaptée selon les tendances générales.

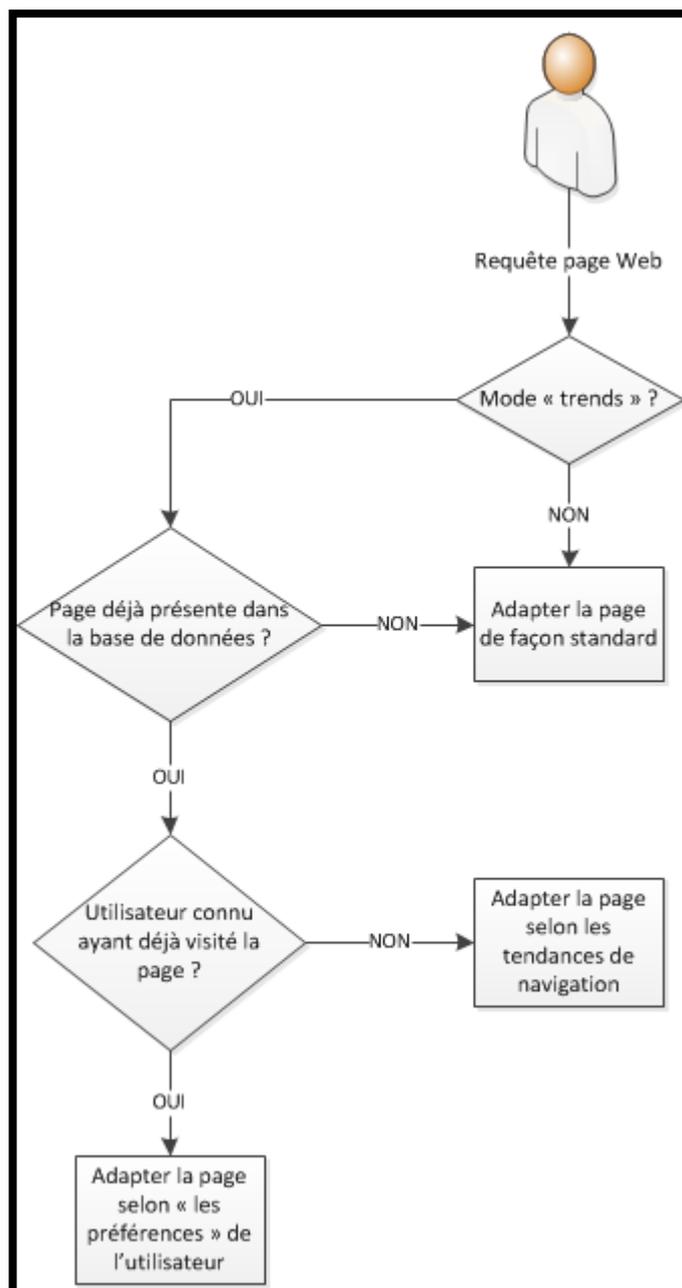


Figure 7 - Processus de détermination du contenu présenté à l'utilisateur

### 3.1.4 Conséquences

Cette solution basée sur les tendances de navigation présente un certain nombre d'avantages :

- Amélioration de l'expérience utilisateur : Le contenu délivré comporte par définition moins d'éléments, ce qui a un impact sur la taille (en termes de mémoire) du document, engendrant une vitesse de réception améliorée. L'expérience globale est

augmentée, étant donné que le contenu est épuré. La présentation de données peut se dérouler dans un plus grand espace, améliorant la lisibilité et évitant toute surcharge qui provoque généralement la confusion chez l'utilisateur.

- Les techniques actuelles restent la base : Les techniques d'adaptation couramment utilisées à l'heure actuelle ne sont pas remises en cause et seront toujours utilisées. Le travail réalisé n'est donc pas perdu. Notre solution peut être vue comme un complément à ce qui existe déjà, une couche supplémentaire offrant des services plus avancés.
- Transparence : Les changements induits par cette solution sont transparents pour les utilisateurs et les auteurs de sites. Le processus est automatisé et ne requiert pas d'efforts de la part des parties concernées. Les auteurs ne doivent pas modifier leur manière de concevoir des documents Web (pas d'ajout de métadonnée, pas de conception spécifique pour mobile, etc). Du côté des utilisateurs, ils doivent uniquement spécifier quel type de version ils désirent recevoir.
- L'utilisateur a le dernier mot : Il reste l'élément central et peut décider ou non de profiter des fonctionnalités proposées.

## 4 Etude de cas

Dans cette étude de cas, nous illustrerons ce que donne de manière théorique notre solution. Pour ce faire, nous partirons de sites Internet réels et poserons à chaque fois une hypothèse concernant les tendances de navigation afin de pouvoir décider quel contenu privilégier pour l’affichage. Nous comparerons dans chaque cas ce que nous obtenons avec les résultats retournés par « Opera mini » et Skweezer, les deux meilleures solutions que nous avons déjà testées précédemment.

Notre développement portera sur 3 catégories de sites, dont le critère de différenciation est la taille. Par là, nous entendons la complexité du site : nombres d’éléments sur une page, structure, nombres de zones différentes, médias (images, vidéos, etc.).

### 4.1 Site de petite taille

Pour appliquer notre solution sur un « petit » site, nous avons sélectionné la page d’accueil du moteur de recherche Google. Cette dernière ne contient en effet pas beaucoup d’éléments et est déjà très épurée point de vue design. Une version optimisée pour mobiles existe mais nous préférons tester la version normale, puisque dans la réalité il y a encore peu de webmasters qui développent une version mobile de qualité.

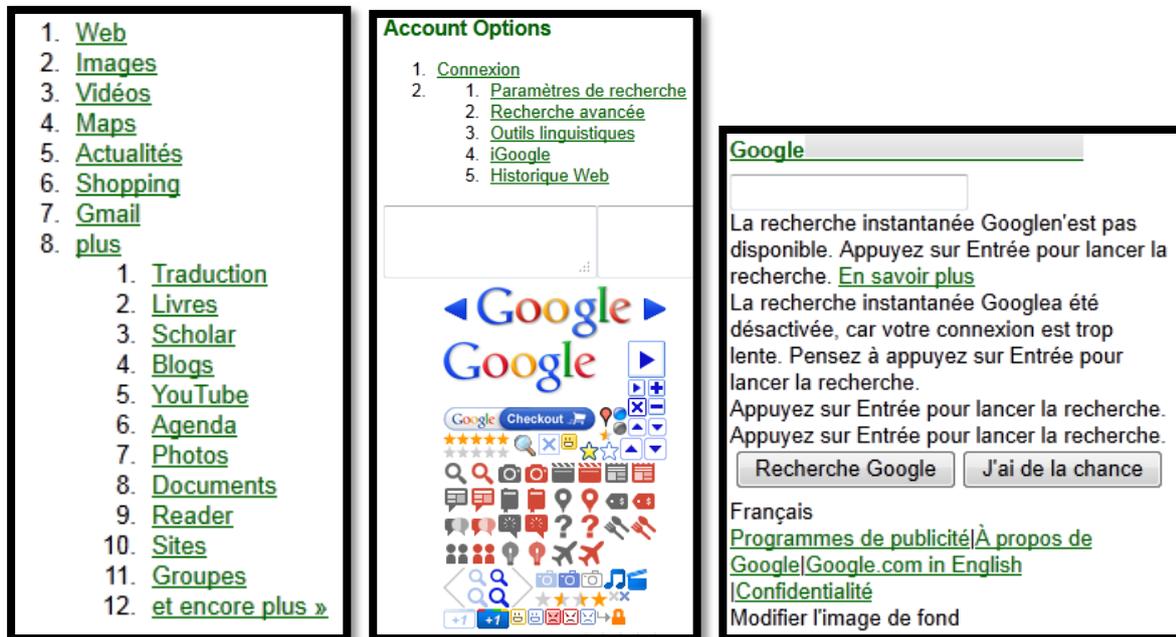
#### 4.1.1 Opera Mini

Le résultat créé par « Opéra mini » à partir de la page non-optimisée de Google est très satisfaisant. Tous les liens de la page originale se retrouvent condensés et centrés sur un espace réduit. Les services additionnels de Google sont disponibles ainsi que le bouton de connexion. Et, bien entendu, la fonction de recherche est également présente. Le seul défaut est la partie du style original et le logo de la firme qui a disparu, mais d’un autre côté, cela allège la taille de la page.



### 4.1.2 Skweezer

Skweezer semble étrangement avoir du mal à adapter correctement la page de Google, alors qu'il retourne d'excellents résultats pour des sites visuellement plus complexes.



Tous les menus ont été déroulés ce qui prend une place considérable et nous oblige à aller plus bas dans la page pour obtenir le formulaire de recherche. Une image composée des différents symboles utilisés sur les différents sites de l'entreprise apparaît et des phrases viennent se glisser entre le formulaire de recherche et les boutons de validation. Cette adaptation ne produit pas une expérience correcte pour l'utilisateur et la version mobile du site développée par Google prend tout son sens dans un cas pareil.

### 4.1.3 Notre solution

En posant l'hypothèse que la très grande majorité des utilisateurs qui se rendent sur cette page le font dans le but d'effectuer une recherche (disons 95% du temps) ou pour profiter d'autres services de Google tels Gmail, etc. (5% du temps), nous obtenons cette version adaptée selon les tendances globales de navigation.

Elle est comparable à celle proposée par « Opéra Mini », mais avec certains liens en moins. Les liens considérés comme les moins utilisés n'ont pas été sélectionnés, ce qui nous a permis de



profiter de l'espace libéré pour « aérer » la présentation générale et offrir un confort accru à l'utilisateur.

## 4.2 Site de taille moyenne

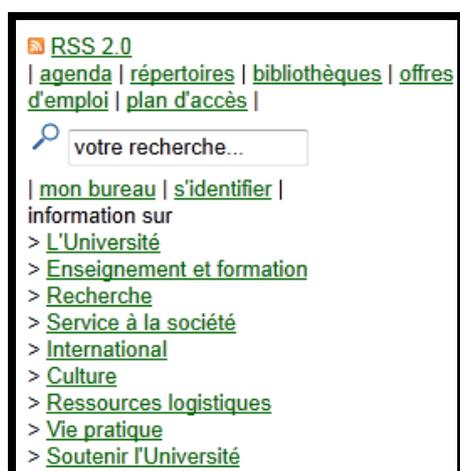
Nous reprenons le site de l'UCL pour l'analyse des résultats sur un site de taille moyenne. Les raisons de ce choix ont déjà été citées lors de la comparaison des différentes solutions pratiques d'adaptation. Cela nous permettra également de reprendre les résultats obtenus avec « Opera Mini » et Skweezer, que nous commenterons nettement moins que dans la partie « Etude critique de l'existant ».

### 4.2.1 Opera Mini

La présentation proposée est remarquable et conserve tout le style du site original. L'utilisateur est déjà familiarisé avec les couleurs et se sentira très à l'aise pour naviguer sur une telle version mobile.



### 4.2.2 Skweezer



La version proposée par Skweezer est moins conviviale mais tout autant fonctionnelle. On constate qu'apparaît exactement la même structure de liens que sur Opéra, mais que le style n'a pas été conservé, mais remplacé par une police uniforme caractéristique de Skweezer.

### 4.2.3 Notre solution

Nous nous mettons cette fois-ci dans la peau d'un étudiant de l'UCL. Posons comme hypothèse qu'en tant qu'étudiant, lorsqu'il se connecte à la page d'accueil de l'UCL c'est pour se connecter à son compte via « mon bureau » ou « s'identifier », afin d'aller voir ses e-mails ou ses résultats d'examens. Notre étudiant utilise également beaucoup le champ de recherche du site et lit de temps en temps les nouvelles en rapport avec l'université.

Dans le cas où cet étudiant choisi d'obtenir une version adaptée à ses propres habitudes de navigation, le système pourrait lui présenter la page suivante :



Les ressources les plus souvent utilisées que sont la recherche et les liens d'identification sont mises en évidence, tandis que la nouvelle la plus récente est affichée en-dessous. Encore une fois, cette version utilise les mêmes principes que ceux d'Opéra mais propose un contenu plus pertinent en fonction des préférences de l'utilisateur.

### 4.3 Site de taille importante

Les sites d'e-commerce ont généralement tendance à être plus complexes que les sites traditionnels, surtout lorsqu'ils proposent une très large gamme de produits très diversifiés. Les pages d'accueil de tels sites sont très vastes et comportent dans ce cas de nombreuses zones et un nombre très élevés de liens qui ont pour but de fournir un accès rapide aux différentes sous-sections du site.

Nous avons sélectionné le site « ldlc.be » comme site relativement complexe par rapport aux deux précédents que l'on a testé. Cette fois encore nous choisissons la page d'accueil comme page de référence.

#### 4.3.1 Opéra mini

La version mobile proposée semble sérieuse en ce qui concerne la première capture d'écran. On y retrouve en haut de l'écran les catégories de tout produit vendu par le site, accompagné d'un champ de recherche qui représente un moyen très rapide pour retrouver un produit déterminé. Des liens de connexion/inscription sont également présents.

Enfin, des boutons « call to action » sont disposés en bas de l'image et permettent à l'utilisateur d'accéder à des sections précises et intéressantes du site.

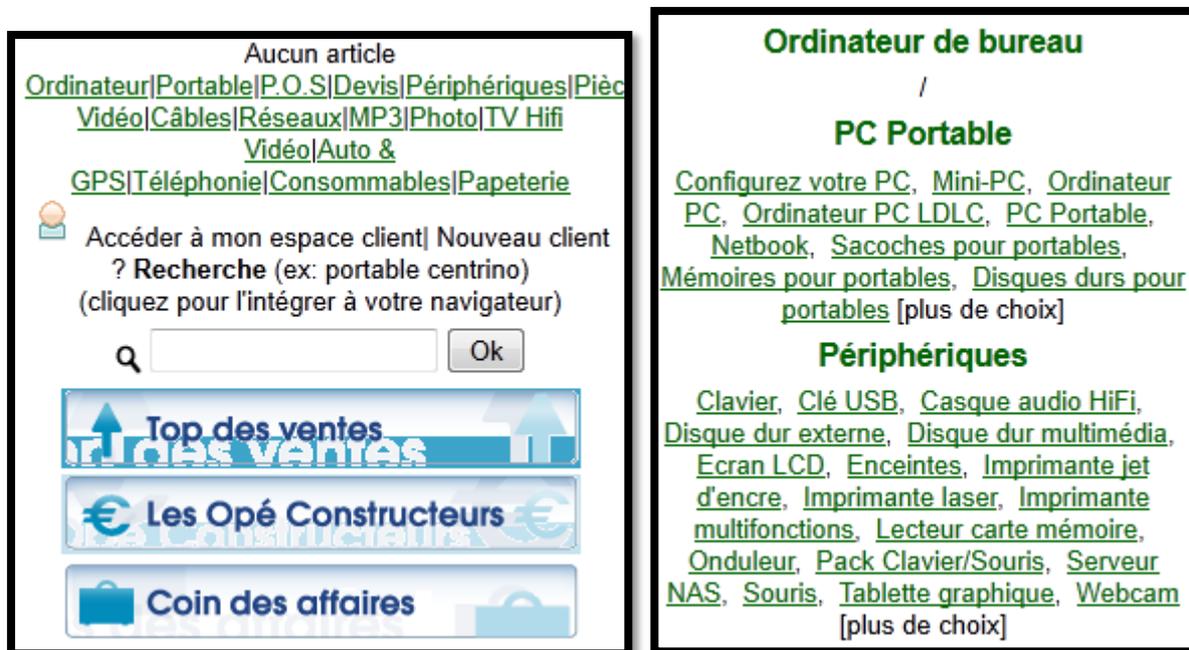


La seconde image montre comment sont gérés les nombreux liens qui pointent vers des sous-catégories et qui offrent à l'utilisateur un accès très rapide à une famille de produits donnée.

On remarquera que cette fois-ci la présentation réalisée par opéra est un peu « brouillon » et l'on peine à s'y retrouver. Les titres de section qui sont normalement mis en valeur sont ici plus petits que le contenu en lui-même !

### 4.3.2 Skweezer

La version Skweezer est presque identique à celle d'Opera mini. Nous pouvons y retrouver la même structure. Cependant la présentation est plus claire, le texte est plus aéré, les titres de sections sont mis en évidence.



Nous constatons néanmoins un inconvénient majeur puisque les liens de connexion/identification ne sont pas fonctionnels.

### 4.3.3 Notre solution

Pour un site de cette importance, qui comporte un nombre très élevé de liens, notre solution semble moins efficace. En effet, il est fort probable qu'aucune tendance claire ne ressorte en temps normal.

Dès lors, on pourra se rabattre sur la navigation personnalisée selon les habitudes de l'utilisateur, mais dans la plupart des cas nous devons procéder de la même manière que les outils traditionnels.

## 5 Conclusion

### 5.1 Résumé

Nous avons au cours de ce travail analysé des techniques théoriques mais également pratiques d'adaptation de contenu web pour des dispositifs mobiles. Parmi les solutions proposées, certaines montrent une grande qualité et s'avèrent être prometteuses pour le futur. A l'heure actuelle, c'est « Opera mini » qui semble être la meilleure offre dans le domaine de l'adaptation de contenu Web.

Nous avons par la suite tenté de proposer une solution alternative à ce qui se fait actuellement, basée sur les tendances de navigation des utilisateurs. Le but de cette solution est de proposer un contenu épuré mais efficace à l'utilisateur final. Les quelques tests théoriques effectués nous montrent que cette solution propose des résultats intéressants.

### 5.2 Limites

Notre solution est susceptible d'être moins efficace, sur des sites jugés plus complexes, où des tendances claires sont moins perceptibles.

De manière globale, l'adaptation du contenu n'est pas une science exacte et aucune solution ne peut affirmer qu'elle est la meilleure sur tous les points. Chacune présente des avantages et inconvénients et il y a toujours marges de progression à réaliser.

Du point de vue de l'état du Web, beaucoup de recommandations et travaux du W3C ne sont pas encore implémentés par la plupart des navigateurs du marché. Cela représente un frein pour le moment mais nous pouvons être sûrs que le Web est sur la voie de standardisation et que les navigateurs vont prendre en compte les recommandations du consortium.

### 5.3 Travaux futurs

Le W3C travaille activement sur le sujet de l'accessibilité via tout type de media et dispose de sa section dédiée sur le site du consortium. La navigation à partir de dispositifs mobiles est en grande expansion et l'on peut être sûr que des solutions de plus en plus développées feront leur apparition au cours des prochaines années.

Avec ses métadonnées, le web sémantique est une option sérieuse et apportera certainement beaucoup de solutions à des problèmes qui se posent actuellement.

Nous pensons également que les éditeurs de sites Internet doivent y mettre du leur et ne pas se reposer entièrement sur les techniques d'adaptation qui sont souvent automatiques et traitent des cas généraux, ce qui peut poser problème pour des sites spécifiques.

Des solutions basées sur la pertinence peuvent répondre d'une manière plus adéquate aux besoins des utilisateurs.

## 6 Bibliographie

1. **World Wide Web Consortium.** Delivery Context Overview for Device Independence. [En ligne] 20 Mars 2006. <http://www.w3.org/TR/2006/NOTE-di-dco-20060320/>.
2. *Adaptation dans les systèmes d'information spatio-temporelle interactifs.* **Moisuc, Bogdan.** Hammamet : s.n., 2006.
3. **World Wide Web Consortium.** Authoring Techniques for Device Independence. [En ligne] 18 février 2004. <http://www.w3.org/TR/di-atdi/>.
4. **World Wide Web Consortium.** Media Queries. W3C. [En ligne] 27 Juillet 2010. <http://www.w3.org/TR/css3-mediaqueries/>.
5. **Alsacrérations.** Les Media Queries CSS3. *Alsacrérations.* [En ligne] 9 Février 2011. <http://www.alsacreations.com/article/lire/930-css3-media-queries.html>.
6. **World Wide Web Consortium.** Synchronized Multimedia Integration Language (SMIL 3.0). W3C. [En ligne] 1 Décembre 2008. <http://www.w3.org/TR/SMIL/>.
7. —. Xforms Implementations. [En ligne] 10 Février 2011. [http://www.w3.org/MarkUp/Forms/wiki/XForms\\_Implementations](http://www.w3.org/MarkUp/Forms/wiki/XForms_Implementations).
8. *SMIL Content Adaptation for Embedded Devices.* **Lelmlouma, Tayeb et Nabil, Layäida.** 2003. Synchronized Multimedia Integration Language European Conference. pp. 12--14.