Étude Bibliographique de Master 2

Spécialité : **AIGLE**

Personnalisation de page web : application à l'amélioration de l'accessibilité du web

par **Franck PETITDEMANGE**

Mars 2014

Sous la direction de **Marianne HUCHARD, Michel MEYNARD, Yoann BONAVERO**

Contents

1 Introduction et motivation

Le World Wide Web (www) est un réseau de ressource. La publication de ces ressources repose sur un langage universellement compréhensible et accepté par tous les ordinateurs : HTML. Historiquement conçut pour faciliter l'échange d'article dans la communauté scientifique. La démocratisation du web a fait radicalement évoluer le contenu d'une page web, sans pour autant que le langage ne suive ces évolutions. Ainsi les auteurs de page web ont détourné les pratiques de conception d'une page de manière anarchique. Ce manque d?homogénéité complique la compréhension du contenu publié sur le www par une machine. Ceci faisant perdre la propriété universelle du web voulu par son créateur Tim Berners Lee :

> "La puissance du Web réside dans son universalité. L'accès à tous, quel que soit son handicap est un aspect essentiel"

Ceci introduit la motivation de ce stage et les problématiques qui en découlent.

Le sujet du stage est : la personnalisation des pages web. L'objectif est de fournir des méthodes et des outils afin d'adapter une page suivant les souhaits d'un lecteur. On s? intéresse à une application pour l'amélioration de l'accessibilité des pages web pour les personnes en situation de handicap visuel.

L'adaptation d'une page web implique notre première problématique : la restructuration d'une page web. On souhaite expérimenter une approche basée sur les méta-modèles. L'idée étant d'extraire la structure d'une page et d'en construire une représentation plus abstraite. Cela doit nous permettre de s'affranchir de la diversité de conception de ces dernières. A partir de cette représentation on veut lui appliquer des transformations, puis générer une nouvelle page conforme aux transformations. L'acquision de ces transformations se fera au travers d'un protocole d'extraction des souhaits du lecteur que nous étudierons par la suite dans le stage.

Dans la section 2 nous précisons l'exploitation que nous souhaitons faire des modèles. Puis en section 3 nous étudions différents modèles de page en perspective de la création d'un meta-modèle. La section 4 présente plus en détail la problématique d'extraction de la structure d'une page web. Elle présente également différentes approches traitant de la reconnaissance de structure.

2 Exploitation des modèles

Dans le cadre du sujet nous nous plaçons dans un contexte de restructuration d'une page. On souhaite construire un modèle d'une page web à partir de cette dernière. Le modèle de cette page est construit depuis un méta-modèle.

Definition. "Un modèle est une description d'une partie d'un système écrit dans un langage formel"[10]

C'est une représentation simplifiée d'une partie d'un système, une abstraction du système étudié suivant un point de vue. Par exemple une carte routière est une abstraction d'un réseau routier, il existe plusieurs types de carte suivant ce que l'on veut étudier (chemin pédestre, chemin routier etc).

> "Pour un observateur A, M est un modèle de l'objet O, si M aide A à répondre aux questions qu'il pose sur O" (Minsky)

La notion de modèle fait référence à un langage formel définissant les éléments conceptuels du modèle : le méta-modèle. La syntaxe et la sémantique du modèle sont conformes à un méta modèle.

Definition. "Un méta-modèle est un modèle qui définit un langage formel pour exprimer un modèle"[10]

Littéralement, c'est un modèle de modèle.

L'avantage d'utiliser une telle construction est de manipuler une représentation indépendante de la diversité de conception des éléments pages. En effet les éléments, ou concepts, récurrents peuvent être structurés ou mises en forme de manière différente d'une page web à l'autre. [16] présente une méthodologie similaire. Les auteurs souhaitent appliquer des traitements homologues sur des systèmes orientés objets implémentés dans différentes plateformes (C++, Java, ADA, etc). Pour cela ils ont réalisé un meta-modèle de système objet dans le but d'effectuer des traitements independamment d'une représentation sous-jacente.

Nous voulons exploiter ce modèle de deux façons différentes. D'une part ce modèle doit nous servir de support pour appliquer un processus de transformation sur une page. Et d'autre part le modèle doit nous servir de support à l'extraction des règles de transformation, que nous ne traiterons pas dans la bibliographie mais par la suite dans le stage.

Definition. "Une transformation est la génération automatique d'un modèle cible depuis un modèle source selon une transformation définie"[10]

Une transformation est donc un ensemble de règles qui décrivent comment un modèle du langage source peut être transformé dans un modèle du langage cible. Plus précisément, elle est la description de comment une ou plusieurs constructions d'un langage source peuvent être transformées dans un langage cible. Plusieurs approches pour la transformation de modèle existent [9].

Suivant le langage du modèle source et du modèle cible, intervient une distinction (fig.1). On parle de transformation endogène (rééchelonnement) lorsque les modèles sont exprimés dans le même langage. On parle de transformation exogène (translation) dans le cas contraire.

Il s'agit de transformation horizontale quand le modèle source et cible possèdent le même niveau d'abstraction. Lorsque ce n'est pas le cas, il s'agit de transformation verticale (fig.2).

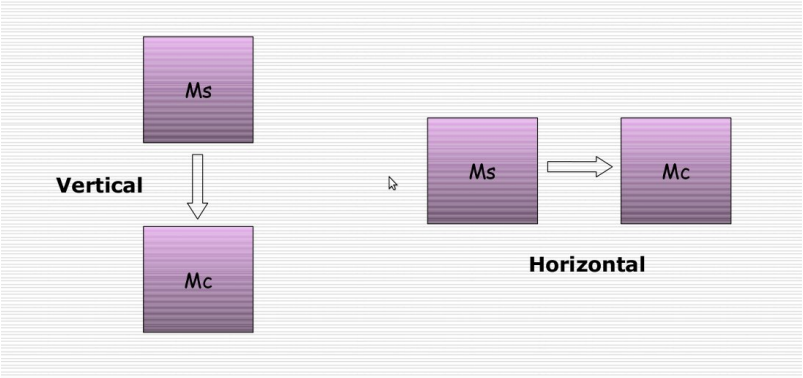

Figure 1: Transformation horizontale et verticale

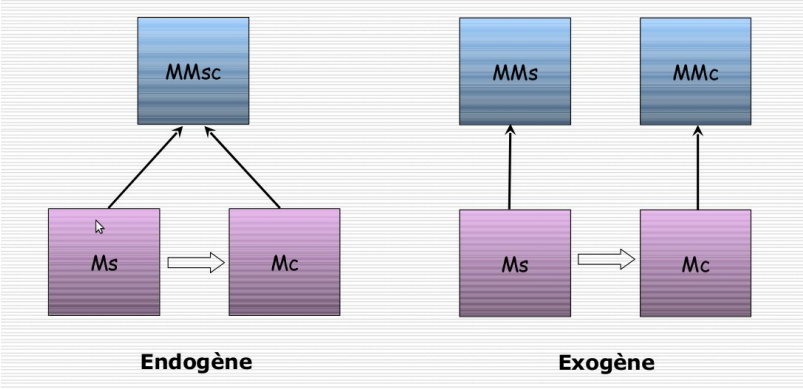

Figure 2: Transformation endogène et exogène

On distingue plusieurs transformations pour notre approche (cf figure 3). Une première est une transformation horizontale et exogène (cf figure 3 transformation 1). Cette étape correspond à la construction d'un modèle source de page depuis une page web. Ce processus est l'objet de la section 4. Le modèle construit est conforme à un possible méta-modèle intermédiaire étudié dans la section 3. Puis une transformation verticale et endogène (cf figure 3 transformation 2). Cette étape sera étudié plus tard dans le stage. Elle intègre dans ces transformations les souhaits de personnalisation d'une page web par d'un utilisateur. L'acquisition de ces transformations fera l'objet d'un protocole d'acquisition des souhaits. Et pour finir un transformation horizontale et exogène (cf figure 3 transformation 3) qui va nous permettre de construire la page web consulté par l'utilisateur suivant ses souhaits.

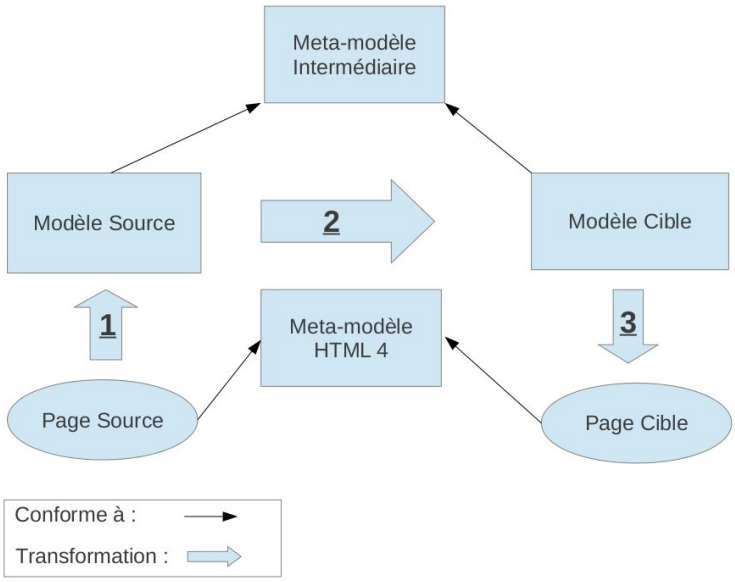

Figure 3: Processus de transformation

3 Modele de page web

3.1 Introduction

Comme expliqué dans la section 2, le méta-modèle est le support à l'expression de transformation. Ainsi les concepts décrits par le méta-modèle doivent comprendre les éléments, ou concepts que l'on souhaite transformer dans une page. Par exemple, si l'on souhaite modifier la couleur du texte de la bar de navigation, on doit pouvoir exprimer dans la transformation le concept de bar de navigation et le concept de couleur.

On s?intéresse à la conception d'un méta-modèle proche de la vision qu'un lecteur peut avoir d'une page. Le but étant d'exprimer des transformations conformes aux souhaits du lecteur. On propose d'étudier et de comparer le langage standard de publication de page sur le web, HTML, dans la norme 4 et 5 mais aussi une taxonomie pour la description d'interface graphique ARIA.

3.2 HTML 4

HTML 4 [5] est un langage permettant la publication de contenu sur le web. C'est le langage standard actuel des pages web. Il permet de structurer le contenu et de lui associer une mise en forme. Le contenu peut être du texte, des images, ou plus généralement du multimédia. Ce contenu est organisé de manière hiérarchique en le découpant en section et sous-section.

Contenu Le contenu principal décrit dans les pages HTML 4 est un contenu textuel. Il peut également contenir du multimédia comme des images, des vidéos et applets (des programmes qui sont automatiquement chargés puis lancés sur la machine de l'utilisateur). L'inclusion de contenu multimédia se fait par l'élément générique : <OBJECT>. Il possède une collection d'attribut prédéfinit qui décrit l'objet inclus dans la page. Le principale étant *type* décrivant le type de contenu des données (e.g figure 4). La valeur de ces attributs n'est pas prédéfinit. Elle est interprété librement par la machine qui charge la page web.

```
<OBJECT data="canyon.png" type="image/png">
        Ceci est une image
</OBJECT>
```

Figure 4: Exemple contenu multimédia

Structuration générique HTML 4 propose un mécanisme générique pour la composition de contenu formant la structure des pages web. Ce mécanisme gravite autour des éléments de type <DIV> leurs identifiants respectives : id et classe.

DIV Signifiant division, est utilisé comme conteneur générique , il peut contenir n'importe quel élément. Il est exploité pour :

- regrouper les éléments pour leur appliquer un style (une mise en forme particulière).
- signaler une section ou une sous-section.

id et class Chaque élément peut se voir attribuer un identifiant ou une classe d'appartenance. *id* assigne un nom à un élément. Ce nom est unique dans le document. *class* au contraire, assigne un ou plusieurs noms de classe à un élément. Un nom de classe peut être partagé par plusieurs instances d'éléments. Les identifiants et les classes sont des suites de caractères quelconque décidées arbitrairement par l'auteur du document.

Les éléments DIV utilisés conjointement avec les attributs id et classe sont au cœur du mécanisme générique de structuration d'un document. DIV permet de diviser le contenu d'un document en section et sous-section (e.g figure 6) pour décrire sa structure. Les éléments <DIV> ayant une sémantique neutre, c'est l'auteur du contenu qui attribut (de manière arbitraire) un nom de *class* ou un *id*. L' *id* ou la *class* est associé à une mise en forme définit à priori. La mise en forme est définit au travers d'un langage : CSS[6] que l'on appel feuille de style. CSS permet d'appliquer un ensemble de règle de style ou un agencement dans l'espace de la page aux éléments. Par exemple, l'auteur peut déclarer une classe "aside" est définir que les éléments appartenant à la classe "aside" doivent être placé sur le côté droit de la page avec un fond blanc.

Ce mécanisme est illustré par la figure 5, les balises <DIV> découpent le contenu par thématique (entête, pied de page, etc). L'auteur associe à chaque <DIV> une *class* ou un *id* auquel s'applique une mise en page et une mise en forme définit par l'auteur dans une feuille de style CSS.

```
<body>
        <div id="header" ></div>
        <div id="navigation_bar"></div>
        <div class="aside"></div>
        <div class="section">
                <div class="article"></div>
                <div class="article"></div>
        </div>
        <div class="aside"></div>
        <div id="footer"></div>
</body>
```

Figure 5: Architecture page web HTML 4

```
<body>
<div class="section" id="elephants-foret" >
        <h1>Les éléphants des forêts</h1>
        <p>Dans cette partie, nous abordons le sujet
moins connu des éléphants des forêts.</p>
        <div class="sous-section" id="habitat-foret" >
        <h2>L'habitat</h2>
        <p>Les éléphants des forêts ne vivent pas
dans les arbres mais au milieu d'eux.</p>
        </div>
</div>
</body>
```

Figure 6: Exemple découpage en section et sous-section

3.3 HTML 5

HTML 5 [8] étend HTML 4 en apportant de nouveaux éléments lexicaux. Ces nouveaux éléments apportent une sémantique standard et explicite à la structure d'une page ainsi que le contenu.

Contenu HTML 5 fournit de nouveaux éléments comme <VIDEO>, <AUDIO> avec un ensemble d'attribut propre à chaque balise (à contrario de l'élément <OBJECT> de HTML 4). Les attributs spécifiques permettent de renseigner l'état d'un élément. Par exemple, la balise <AUDIO> possède un attribut spécifique *muted* indiquant si le son de l'élément audio est coupé ou non.

Structuration Les nouveaux éléments de HTML 5 spécifient donc une sémantique standard :

- SECTION : représente une section générique dans un document, c'est à dire un regroupement de contenu par thématique.
- ARTICLE : représente un contenu autonome dans une page, facilite l'inclusion de plusieurs sous documents.
- NAV : représente une section de liens vers d'autres pages ou des fragments de cette page.
- ASIDE : représente une section de la page dont le contenu est indirectement lié à ce qui l'entoure et qui pourrait être séparé de cet environnement.
- HEADER : représente un groupe d'introduction ou une aide à la navigation. Il peut contenir des éléments de titre, mais aussi d'autres éléments tels qu'un logo, un formulaire de recherche, etc.
- FOOTER : représente le pied de page, ou de la section, ou de la racine de sectionnement la plus proche.

La figure 7 montre un découpage explicite de la structure avec HTML 5 à contrario du découpage implicite de HTML 4 montré dans la figure 5.

```
<body>
        <header></header>
        <nav></nav>
        <section>
                <article></article>
                <article></article>
        </section>
        <aside></aside>
        <footer></footer>
</body>
```

Figure 7: Exemple d'attribution de rôle

3.4 ARIA

ARIA (Acessible Rich Internet Application) [7] est la spécification d'une ontologie décrivant une interface graphique. Elle fournit des informations sur la structuration d'un document et plus généralement décrit les éléments qui composent une interface au moyen d'un ensemble de rôles, d'états et de propriétés .

Rôle Les rôles permettent d'identifier la fonction de chaque élément d'une interface. Ils sont regroupés en trois catégories :

- Widget Roles : référence un ensemble de widget prédéfinis (alertdialog, button, slider, scrollbar, menu, etc)
- Document Structure Roles : décrit les structures qui organisent un document (article, definition, entête, ect)
- Landmark Roles : décrit les régions principales d'une interface graphique (main, navigation, search, etc)

Etats et propriétés ARIA prend en compte l'aspect dynamique et interactif des éléments d'une interface. Elle permet d'associer des états et des propriétés aux éléments d'une interface. Un état est une configuration unique d'un objet. Par exemple, on peut définir l'état d'un bouton par l'état *aria-checked* qui peut prendre trois propriétés suivant l'interaction avec l'utilisateur : *true* - *false* - *mixed* . Dans le cas d'une checkbox, *true* indique si la checkbox est coché, *false* si elle ne l'ai pas et *mixed* dans le cas d'un ensemble de checkbox indique que certaine sont cochés.

Aria prévoit même un système d'annotation pour les objets ayant des comportements asynchrones. Par exemple, on peut annoter qu'un élément se met à jour de manière autonome.

3.5 Discussion

L'étude ci-dessus présente de manière sommaire les concepts de langage de HTML 4, 5 et ARIA. Au regard de notre problématique, nous pouvons voir quelques perspectives à la conception d'un méta-modèle.

HTML 4 fournit une sémantique riche pour décrire les éléments textuels d'une page et ses hyperliens. Cependant le mécanisme de structuration fournit est trop générique et ambiguë. En effet, il n'est pas possible de délimiter explicitement les éléments d'une page par thématique. Le mécanisme d'imbrication hiérarchise le contenu alors qu' un élément n'est pas forcement sous-section de l'élément qui l'imbrique. Un page web contient des éléments qui ne sont pas en rapport avec le contenu d'un document, mais plutôt avec le site web (e.g menu de navigation, un logo, etc).

HTML 5 semble combler ces limites en apportant de nouveaux éléments lexicaux amenant une sémantique plus riche et surtout standard sur la structure d'une page et la nature du contenu. Elle permet donc de modéliser les limites de chaque partie d'une page suivant ça thématique (<SECTION>). Elle permet de décrire un contenu non-linéaire (<ASIDE>). Et introduit également des balises pour la description d'élément propre à une page web comme des menus de navigation, bannière, logo, etc.

La norme ARIA intègre plus d?éléments pour décrire une page web et notamment une description du comportement des éléments d'une page web particulièrement utiles avec le développement des éléments graphiques appelés widget.

Un méta-modèle basé sur la syntaxe de HTML 5 semble être un compromis intéressant entre HTML 4 et ARIA. HTML 4 est peu adapté à l'expression de transformation reflétant l'intention d'un lecteur et ARIA est trop riche au regard de nos besoins immédiats. Les spécifications de HTML 4 sont disponible sous forme de DTD[1]. Une DTD est document décrivant la grammaire d'un langage. Il est simple d'en construire un méta-modèle. HTML 5 n'est pas encore au rang de standard, aucune DTD n'est encore disponible, seul une définition exhaustive des éléments lexicaux et leurs fonctions sont fournit par la W3C. Il n'y pas de grammaire clairement spécifié. La tâche de réalisation d'un méta-modèle semble complexe pour HTML 5. Aria fournit lui un modèle UML[2]. UML est un langage de modélisation est un langage de modélisation graphique à base de pictogrammes[3]. Ce modèle UML est pour nous un méta-modèle directement exploitable. Ainsi nous pourront utiliser le modèle fournit par ARIA pour la construction d'un méta-modèle. Ce dernier serait restreint aux éléments de description d'un document (*Document Structure Roles*) et (*Landmark Roles*). Suivant l'évolution de nos besoins de transformation nous pourrions facilement intégrer d'autres aspects, notamment les widget et comportements asynchrones des éléments d'une page. Ce méta-modéle intégrera les éléments de HTML 4, en particuliers ceux décrivant le contenu textuel non modélisé avec ARIA (e.g paragraphe).

4 Extraction de structure

4.1 Introduction

L'extraction intègre le processus de transformation 1 de la figure 3 . On veut extraire les éléments de la structure d'un page qui correspondent aux différentes briques conformes aux éléments d'un méta-modèle de page définit à priori. La problématique est due au langage de conception des pages web qui manque de sémantique, rendant un processus d'extraction automatique difficile.

En effet, La conception des pages web s'articule autour : d'un langage pour décrire la structure du document (HTML) et d'un langage pour décrire la mise en forme du document (CSS). Les pages sont constituées d'éléments hétérogènes : une page est constituée d'un ou plusieurs contenu principal, d'un menu de navigation, de publicité, etc... Chacun de ces éléments représentent une sous-structure de la page. Lorsqu'on regarde une page web depuis un navigateur, on constate que ces éléments sont structurés de façon sémantique, ils sont organisés selon leur sens. La difficulté dans la tâche d'extraction de la structure d'une page est dûe au manque d'expressivité du HTML. La norme actuelle de HTML (HTML 4), ne fournit pas de moyen de délimiter les éléments du document en fonction de leur sémantique. Par exemple, on ne peut pas délimiter de manière explicite la structure d'un menu dans une page avec ce langage. Le constat est que l'information de la structure d'une page apparaît principalement dans la mise en page. La structure d'une page est explicitée à travers l'utilisation de police, de couleur ou plus généralement d'éléments visuels pour caractériser les contenus qui ont la même signification.

Une approches étudiées dans la tâche d'extraction de la structure est un processus de segmentation. Ce dernier permet de découper une page en région, on peut ainsi délimiter le contenu d'une page de façon explicite. Nous avons étudié également des approches de reconnaissance de structure syntaxique et fonctionnelle.

[1] Document Type Definition
[2] Unified Modeling Language
[3] wikipédia

4.2 Segmentation de contenu

Comme présenté dans la section 3.2 HTML 4 fournit une sémantique ambiguë avec la balise <DIV>. On ne sait pas si elle délimite une section, sous-section ou si elle est utilisé pour appliquer un style. Le sens n'en est connu que par le concepteur de la page. La compréhension de la structure d'une page est alors implicite. Les auteurs de l'article "Extracting content structure for web pages based on visual representation"[2] propose une approche pour rendre explicite la structure de la page.

4.2.1 Approche segmentation visuelle

L'approche proposée par les auteurs [2] présente un algorithme de partitionnement basé sur les éléments de mise en forme des pages web. Le partitionnement extrait une structure qui regroupe les éléments d'une page sémantiquement proche en bloc (e.g figure 8). Le postulat est que les éléments d'une page possédant des caractéristiques de mise en forme proche, tels que la police, la couleur, la taille, sont sémantiquement proches.

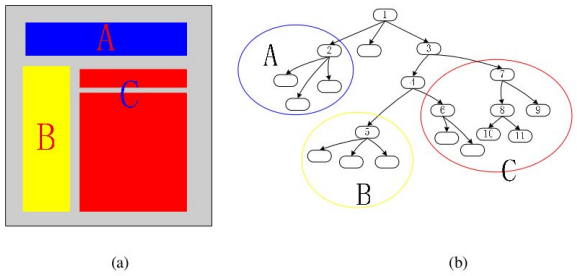

Figure 8: Exemple de partionnement, (a) page (b) DOM de la page

L'algorithme exploite le DOM[4] de la page web. Le DOM est une API[5] pour les documents HTML (ou plus généralement XML) . Il fournit une représentation arborescente d'un document et les moyens d'accéder à son contenu et sa mise en forme.

Le processus de segmentation, figure 9, se décompose en trois phases : un processus d'extraction de blocs, un processus de détection de séparateur et un processus de reconstruction.

Le processus d?extraction détecte les éléments du niveau courant du DOM susceptible de former un contenu cohérent. Cette détection repose sur des séparateurs explicites : on sait que certains éléments délimitent le contour d'un contenu (par exemple les balises <DIV>). Mais également sur une fonction de distance visuelle comparant les n?uds parents et frères du n?ud courant : une balise <DIV> a de grande chance de délimiter un contenu sémantiquement différent du n?ud parent si la couleur de fond est différente de celle de ce dernier. Pour chaque n?ud, l'algorithme vérifie s'il forme un bloc ou non. Si oui, il associe un degré de cohérence au bloc. Ce degré de cohérence est un indicateur de l'importance sémantique du bloc. Si non, il est appliqué le même processus aux enfants du n?ud. Quand tous les n?uds du bloc courant sont extraits, ils sont mis dans un pool.

Des séparateurs entre les blocs sont ensuite détectés. L'algorithme dététecte ici des séparateurs implicites, c'est à dire n' apparaissant pas dans la structure HTML. Les séparateurs implicites

[4]Document Object Model
[5]Application Programming Interface

sont les espaces entre les blocs d'un pool. Un poids est attribué à chaque séparateur suivant son importance (par exemple, plus l'espacement entre deux blocs est grand, plus le poids sera élevé). Ce poids est un indicateur de différence sémantique entre les blocs adjacents. Plus le poids du séparateur est élevé entre deux blocs, plus leur contenu sera sémantiquement éloigné.

Une construction hiérarchique des blocs est crée. Cette construction hiérarchique repose sur le degré de cohérence attribué à chaque bloc.

Pour chaque nouveau bloc de la structure hiérarchique construite, l'algorithme teste le degré de cohérence attribué par rapport à un seuil de cohérence défini. Ce seuil est défini suivant la granularité de la structure que l'on veut en sortie de l'algorithme. Si le degré de cohérence n'est pas supérieur au seuil de cohérence, le bloc est de nouveau proportionné. La structure finale est construite après que tous les blocs soient traités.

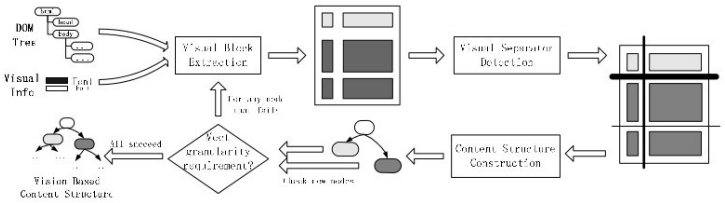

Figure 9: Algorithme de segmentation

4.3 Compréhension de contenu

La segmentation du contenu n'est pas suffisante dans le processus de transformation. Pour construire notre modèle de page, nous avons besoin de faire correspondre les éléments d'une page web aux éléments du méta-modèle intermédiaire. Comme soulevé dans la section 3.2 la sémantique des éléments de structuration est neutre. On ne connaît pas le rôle des conteneurs génériques <DIV>. On ne peut pas faire correspondre les éléments d'une page directement avec les éléments de nôtres méta-modèle.

On propose d'étudier plusieurs approches pour reconnaître les sous-structures composant une page. Une première approche est une analyse syntaxique. On s?intéresse aux processus exploités en bio-informatique pour isoler des structures dans des séquences génomiques et déterminer quelles sont leurs fonctions. On s'intéresse également à une analyse fonctionnelle des structures en s?intéressant à leurs propriétés.

4.3.1 Analyse syntaxique

Similarité de séquence L'un des postulats de base en bio-informatique est qu'une séquence génomique similaire donne une protéine présentant la même fonction[1]. En d'autre terme, des séquences ayant des similitudes (syntaxiques) est un signe de proximité fonctionnelle. Appliqué à une page web, cela signifie que par exemple la structure syntaxique d'un menu d'une page web A est similaire à la structure syntaxique d'un menu d'une page web B. Il semble cohérent d'appliquer ce postulat à une page web. En effet, la publication de contenu sur le web se standardise par l?intermédiaire des scripts [17]. Ceux-ci génèrent des pages de manière automatique et standardisée

(Wordpress, joomla, etc). En construisant une séquence représentative d'un élément de notre méta-modèle, il devrait être possible de déterminer la fonction d'un fragment de page web. Pour déterminer la similarité de deux séquences, les bio-informaticiens utilisent le concept d'alignement[14].

Definition. L' alignement est la mise en correspondance de deux séquences. Soit deux séquences X1:n et Y1:m dont la valeur est définie dans le même alphabet fini Λ. Un alignement est une correspondance entre les lettres de la première séquence et celles de la deuxième, sans en changer l'ordre, et en autorisant éventuellement des « trous ».

```
G A A T C _ T G A C
C A _ _ C G T _ A _
```

Figure 10: Alignement possible des séquences X1=GAATCTGAC,Y1=CACGTA

La mise en correspondance repose sur trois types d'opérations élémentaires : la substitution, l'insertion, délétion. Plusieurs combinaisons d'alignements existent. Lorsque l'on souhaite comparer la similitude de deux séquences, la meilleure solution est celle qui minimise le nombre d'opérations d'insertions et de délétions.

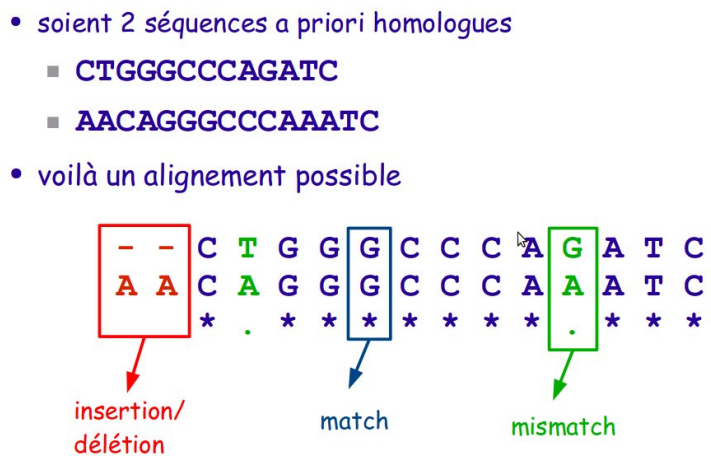

Figure 11: opérations d'alignements

Il existe deux types d'alignements : local et global.

L'alignement global L'alignement global est conçu pour comparer des séquences sur toute leur longueur. Une méthode optimale pour trouver un alignement global maximal de chaîne de caractères est l'algorithme de Needleman-Wunsch [11].

L'alignement local L'alignement local est conçu pour rechercher dans la séquence A des régions semblables à la séquence B (ou à des parties de la séquence B). Une méthode pour trouver un alignement local maximal de chaîne de caractères est l'algorithme de Smith & Waterman [13].

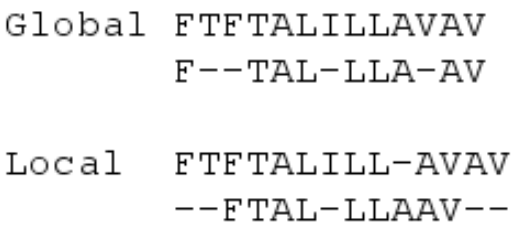

Figure 12: Comparaison sequence d'alignement globale et locale

Comme on le voit dans la figure 12 l'alignement global tente d'aligner les séquences sur toute leur longueur, tandis que l'alignement local se focalise sur les zones de forte homologie. L'alignement global est adapté à la comparaison de deux fragments de taille approximativement égales. Alors que l'alignement local, permet de faire correspondre un fragment dans une plus grande séquence.

Similarité d'arbre Une autre approche dans l'étude de similarité de structure est la comparaison d'arbre [12]. Cette approche Cette approche consiste à trouver la plus petite ou la moins coûteuse séquence d'opération d'édition (substitution, suppression et insertion) qui permet la transformation d'un arbre vers un autre.

Notons Λ un n?ud vide. Une opération d'édition est écrite $b \to c$, où b et c sont soit un n?ud, soit Λ .

- $b \to c$ est une opération de substitution si $b \neq \Lambda$ et $c \neq \Lambda$,
- une opération de suppression si $b \neq \Lambda \doteq c$,
- une opération d'insertion si $b = \Lambda \neq c$

Pour exprimer une séquence d'opération élémentaire qui transforme l'arbre, on utilise le concept de mapping, introduit [15]. Un mapping établie une correspondance un-à-un entre les n?uds de deux arbres ordonnés et qui préservent l'ordre des n?uds.

Definition. Un Mapping M de l'arbre T1 vers l'arbre T2 est un ensemble de paire ordonnée d'entier (i, j), $1 \leqslant i \leqslant n1$, $1 \leq j \leq n2$, satisfaisant les conditions suivantes, pour tous $(i1, j1), (i2, j2) \in M$:

- i1=i2 si et seulement si, j1=j2 (one-to-one condition);
- t1[i1] est à droite de t1[i2], si est seulement si, t2[j1] est à droite de t2[j2] (préservation de l'ordre des n?uds frères);
- t1[i1] est un ancêtre de t1[i2] si et seulement si, t2[j1] est un ancêtre de t2[j2] (préservation de l'ordre des ancêtres);

Definition. Soit M un mapping entre les arbres T1 et T2 décrivant des opérations de modification. S est l'ensemble de pair $(i, j) \in M$, D l'ensemble des n?uds T1[i] n'ayant pas de paire $(i, j) \in M$, et I l'ensemble des n?uds T2[j] n'ayant pas de paire $(i, j) \in M$. Le coût du mapping est donné par $|S|p + |I|q + |D|r|$, où p est le coût des substitutions non identiques, q est le coût des insertions (1), r est le coût d'une suppression (1), le coût des substitutions identiques est 0.

Pour connaître la similarité entre deux structures, on veut calculer une distance d'alignement. C'est à dire trouver le coût minimum du mapping pour que T1 et T2 soient isomorphes. KUO-CHUNG TAI [15] propose un algorithme de programmation dynamique pour résoudre la question de distance d'arbre en temps séquentiel $O(|T1| \times |T2| \times min(depth(T1), leaves(T1)) \times min(depth(T2), leaves(T2)))$.

4.3.2 Analyse fonctionnelle

L'article "Function-based object model towards website adaptation" [4] propose une analyse des fonctions d'un élément d'une page. Cette approche repose sur une segmentation du contenu d'une page à priori pour identifier chaque élément. Cette segmentation est similaire celle de la section 4.2.1. Les éléments isolés par cette segmentation sont vu comme des objets possédant des propriétés (type d'agencement, nombre d'hyperlien, etc). Les auteurs proposent un modèle : Function-based object. Ce modèle est une classification de chaque élément d'une page. Cette classification ce fait sur la valeur des propriétés de chaque objet. Les principales propriétés spécifiés sont basées sur la présentation (type de média, agencement, etc), les hyperliens (le nombre d'hyperlien? vers où pointent ils?), l'interaction avec l'utilisateur (uniquement affichage, soumission de contenu?). L'extraction des propriétés fonctionnelles sont faites principalement par analyse de la nature des balises et les attributs associés aux balises. Suivant la valeur des propriétés, les objets sont rangés dans différentes catégories : objet de type informatif, de navigation, d?interaction, de décoration, etc. L'analyse de la catégorie d'un objet se fait au moyen d'un arbre de décision (e.g figure 13).

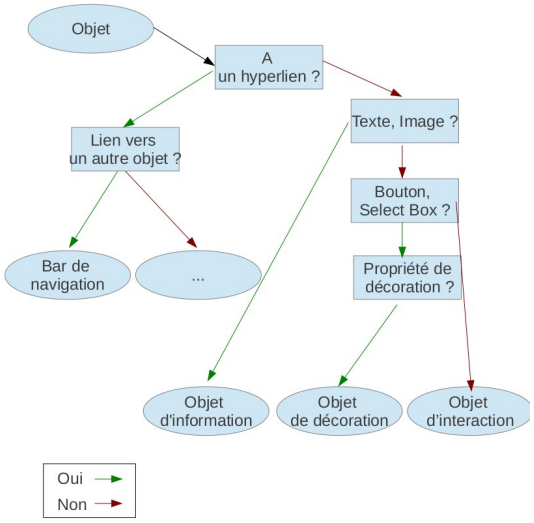

Figure 13: Exemple d'arbre de décision basique

4.4 Discution

L'alignement de séquence locale peut sembler une bonne approche. On peut comparer directement une séquence du méta-modèle à la page pour trouver la zone qui correspond. Cependant le format des structures de données des pages web pose problème. C'est un format de donnée semi-structuré [18]. Il n'y a pas structure formelle pour décrire un type de donnée. Par exemple un menu peut être écrit avec différentes syntaxes (e.g figure 14). Cela complexifie une solution basé sur une analyse syntaxique. En particulier avec une approche basé sur la similarité de séquence. Une solution mise en ?uvre par [3] est une translation des éléments d'une page vers une représentation plus abstraite. Suivant leur classe d'appartenance les balises sont remplacées par un type plus générique (e.g figure 15), par exemple les éléments <UL>, <OL>, <DL> sont différents type de liste, ils seraient remplacés par un type plus générique <LISTS>.

(Similarité d'arbre ??)

L'approche par analyse fonctionnelle (section 4.3.2) semble limiter les inconvénients de l'analyse syntaxique. D'après les résultats de l'expérimentation faite dans l'article, l'approche fonctionne bien. Les éléments de la page sont bien identifiés conformément au modèle définit. Cependant le modèle est rustique et ne couvre pas tous nos besoins.

```
<div class='menu'>
        <ol><a href='#'><span>Home</span></a></ol>
        <ol><a href='#'><span>About</span></a></ol>
        <ol><a href='#'><span>Services</span></a></ol>
        <ol><a href='#'><span>Contact</span></a></ol>
</div>
```

```
<div class='menu'>
<ul>
        <li><a href='#'>Home</a></li>
        <li><a href='#'>About</a></li>
        <li><a href='#'>Services</a></li>
        <li><a href='#'>Contact</a></li>
</ul>
</div>
```

Figure 14: Exemple de conception de menu

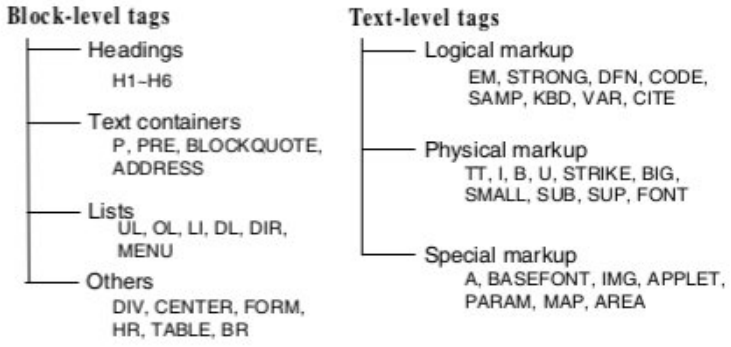

Figure 15: Classification balise

5 Conclusion et futurs travaux

Cette étude bibliographique nous à permis de cerner les avantages d'un processus dirigé par les modèles.

Au travers de l'étude d'un possible support à la réalisation d'un méta-modèle, nous avons synthétisé les principaux concepts des langages de publication sur le web. Cette partie était nécessaire à la compréhension des langages de publication de contenu sur le web et leur futur évolution. Ce travail à également été l'amorce pour cerner la difficulté du traitement compréhension automatique du contenu dans cet environnement. Nous en avons isolé les principales difficultés. Ceci nous à permis d'étudier des solutions existantes et explorer d'autres pistes possibles à l'extraction de structure d?intérêt.

La suite des travaux vont se diriger vers la réalisation d'un méta-modèle, puis le processus de transformation d'une page web pour qu'elle soit conforme à ce méta-modèle. Et pour finir l'étude et la réalisation d'un protocole d'acquisition des souhaits de personnalisation d'une page. Ce protocole présente un grand intérêt pour nous puisque c'est grâce à cette partie que nous souhaitons rendre le web plus accessible aux personnes en situation de handicap. En leur mettant à disposition un outil capable d'adapter le contenu d'une page web de façon personnalisé.

References

[1] Kaizhong Zhang Bruc A.Shapiro. Comparing multiple rna secondary structures using tree comparisons. *CABIOS*.

[2] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Extracting content structure for web pages based on visual representation. In *Web Technologies and Applications*, pages 406–417. Springer, 2003.

[3] Chia-Hui Chang and Shao-Chen Lui. Iepad: information extraction based on pattern discovery. In *Proceedings of the 10th international conference on World Wide Web*, pages 681–688. ACM, 2001.

[4] Jinlin Chen, Baoyao Zhou, Jin Shi, Hongjiang Zhang, and Qiu Fengwu. Function-based object model towards website adaptation. In *Proceedings of the 10th international conference on World Wide Web*, pages 587–596. ACM, 2001.

[5] World Wide Web Consortium et al. Html 4.01 specification. http://www.w3.org/TR/REC-html40/, 1999.

[6] World Wide Web Consortium et al. Cascading style sheets. http://www.w3.org/WAI/intro/aria, 2010.

[7] World Wide Web Consortium et al. Accessible rich internet applications 1.0. http://www.w3.org/Style/CSS/, 2014.

[8] World Wide Web Consortium et al. Html 5 specification. http://www.w3.org/TR/html5/, 2014.

[9] Krzysztof Czarnecki and Simon Helsen. Classification of model transformation approaches. In *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*, volume 45, pages 1–17, 2003.

[10] Anneke G Kleppe, Jos B Warmer, and Wim Bast. Mda explained: the model driven architecture: practice and promise. 2003.

[11] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[12] Bruce A Shapiro and Kaizhong Zhang. Comparing multiple rna secondary structures using tree comparisons. *Computer applications in the biosciences: CABIOS*, 6(4):309–318, 1990.

[13] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[14] Eric Sobel and Hugo M Martinez. A multiple sequence alignment program. *Nucleic acids research*, 14(1):363–374, 1986.

[15] Kuo-Chung Tai. The tree-to-tree correction problem. *Journal of the ACM (JACM)*, 26(3):422–433, 1979.

[16] Sander Tichelaar, Stéphane Ducasse, and Serge Demeyer. Famix and xmi. In *Reverse Engineering, 2000. Proceedings. Seventh Working Conference on*, pages 296–298. IEEE, 2000.

[17] Karane Vieira, Altigran S da Silva, Nick Pinto, Edleno S de Moura, Joao MB Cavalcanti, and Juliana Freire. A fast and robust method for web page template detection and removal. 2006.

[18] Guillaume Wisniewski, Francis Maes, and Ludovic Denoyer. Modèle probabiliste pour l'extraction de structures dans les documents web. *Document numérique*, 10(1):89–107, 2007.