Académie de Montpellier
Université Montpellier II
Sciences et Techniques du Languedoc

MÉMOIRE DE

STAGE RECHERCHE MASTER M2

effectuée au Laboratoire d'Informatique de Robotique
et de Micro-électronique de Montpellier

Spécialité : **AIGLE**

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

par **franck PETITDEMANGE**

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

Table des matières

1 Extraction de contenu

Les processus de d'extraction de contenu recherche sont un support à la classification de document. Les pages web possèdent un contenu informatif que l'on souhaite classifier. Une page web est constitué d'un contenu hétérogène et pas représentatif du contenu de la page. Par exemple un menu de navigation ou la bannière d'une page ne sont pas des informations utiles pour la classification d'une page. On souhaite récuperer le contenu informatif de la page, qui apporte une information. C'est la motivation des différentes approches proposé ci-dessous. Les approches ci-dessous propose des méthodes afin d'identifier les différentes structure logique dans la page et d'inféré des certaines propriétés pour en attribuer une sémantique. La majorité des processus décrit vise à identifier dans une page web le contenu qui décrit une page dans un but d'indexation. Ces processus souhaitent éliminer le *bruit* de ce processus, c'est à dire ne pas prendre en compte les éléments qui ne définissent pas le contenu d'une page d'un point de vue informatique. Par exemple, les menus, une bannière, des pieds de pages.

1.1 Segmentation par zone

Les auteurs de [6] proposent dans leur papier une segmentation des pages web par zone. Cette approche représente les objets HTML par leurs coordonnées cartésiennes et la zone dans laquelle ils sont localisés. L'objectif étant d'améliorer le traitement des informations d'une page web. Dans le contexte de classification de page web, les mots définissants le mieux une page web sont les mots qui se rapprochent le plus de centre le page. L'hypothèse est fondée sur le faite que dans un but d'ergonomie les concepteurs de page web suivent un même modèle de présentation. Les auteurs partitionnent une page suivant ce modèle de présentation en se basant sur une observation empirique de la conception des pages (*cf. figure*1) :

- une entête correspondant à l'emplacement de la bannière d'une page (H),
- un pied de page (F)
- une marge latérale gauche contenant des menus (LF)
- une marge latérale droite contenant des menus (LR)
- du centre de la page encapsulant le contenu principal de la page ((C))

La figure 1 modélise le partitionnement suggéré par les auteurs avec *W1* et *W2* définissant respectivement *LF* et *LR*. Chacune d'elle représente 30% de la largeur de la page. *H1* et *H2* définissent H et F avec un hauteur de 200px pour H et 150px pour F. La méthode propose une représentation des objets HTML dans un système de coordonnées cartésiennes. Les coordonnées sont calculées au moyen d'un moteur de rendu graphique. Il est à noter que tel qu'il est décrit dans l'article, le moteur de rendu graphique ne prend pas en compte la surcouche CSS. En se basant sur ce modèle de représentation, des heuristiques sont déterminées afin de segmenter la page suivant le partitionnement de page proposé.

Heuristique 1 : H se compose des noeuds étiquetés par TABLE est complètement inclus dans la zone *H1*

Heuristique 2 : LM est constitué de tous les noeuds étiquetés par TABLE ou TD qui ne n'appartiennent pas H et sont strictement inclus dans *W1*

Heuristique 3 : RM est constitué de tous les noeuds étiquetés par TABLE ou TD qui ne n'appartiennent pas la H, LM est sont strictement inclus dans *W2*

Heuristique 4 : F est constitué de tous les noeuds étiquétés par table qui ne sont pas contenu dans H, LM, RM et qui sont strictement inclus dans *H2*

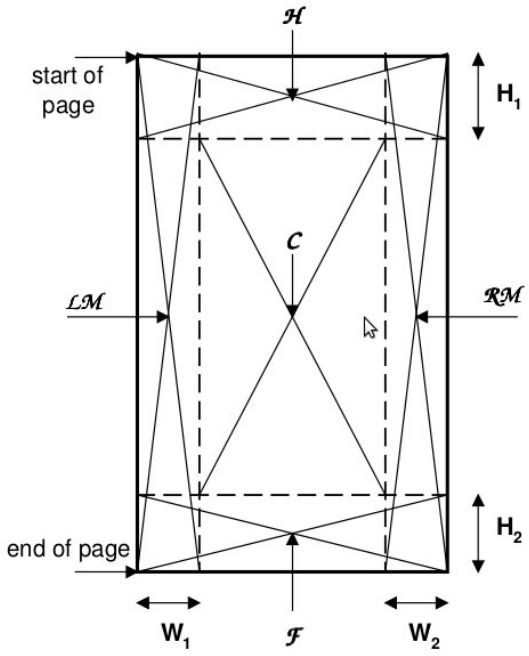

FIGURE 1 – Position des zones d'intérêts dans une page

Heuristique 5 : C est constitué de tous les noeuds qui n'appartiennent pas à H, LM, RM, F

Les heuristiques se basent sur essentiellement sur les étiquettes TABLE qui sont exploités pour la mise en forme des objets HTML dans la page. Cette méthodologie de conception n'est plus utilisé. Ce sont les éléments étiquetés par DIV qui remplissent ce rôle d'agencement des éléments dans les pages.

1.2 Segmentation densitométrique

Pour l'identification des différentes structures dans la page, les auteurs se basent sur l'analyse de la répartition de la densité textuelle dans une page. L'hypothèse des auteurs est que la densité des différents blocs de textes du DOM est une propriété qui suffisante pour segmenter la page en segment représentatifs des différentes structures logiques.

La première étape consiste à identifier les différents segments textuelle de la page. Les auteurs ne prennent pas en compte la sémantique des balises HTML dans le découpage. Le découpage se fait sur l'analyse des changement dans le flux de texte. Par exemple si le flux indique passe d'une sequence de phrase courte à une sequence de phrase longue, cela signale un nouveau segment. Typique si on analyse une sequence de texte courte comme *Accueille, New, Contact* et une sequence de texte longue va donner naissance à deux nouveaux segments. Pour chaque segment (bloc) identifier par le processus ci-dessous. Les auteurs vont calculer une densité textuelle. Cette propriété correspond au ratio ci-dessous :

$$p(b_x) = \frac{NombreDeMotsDansb_x}{NombreDeLignesDansb_x}$$

Les jetons correspondent à une séquence contingente de caractères n'étant pas des caractères d'espacement.Le nombre de ligne est calculé par division du nombre total de caractères dans les blocs par 80. 80 étant une valeur fixée par les auteurs, elle correspond à la taille d'une phrase moyenne (dans les pays anglophones).

Un seconde étape, permettant de trouver les blocs de la page, consiste à fusionner les segments voisins en comparant leur densité textuelle La décision de fusionner les blocs est prise en comparant la densité textuelle des blocs adjacents, en se basant sur la fonction définit ci-dessous :

$$\Delta p(b_x, b_y) = \frac{|p(b_x) - p(b_y)|}{max(p(b_x), p(b_y))}$$

Si la fonction delta est en dessous d'un certain seuil $Smax$ les blocs adjacents sont fusionnés. La fusion de segment consiste ici à joindre les lignes des blocs x et y en un nouveau bloc z. La fonction boucle tant qu'elle trouve des blocs adjacents qui ne respectent pas un seuil définit à priori.

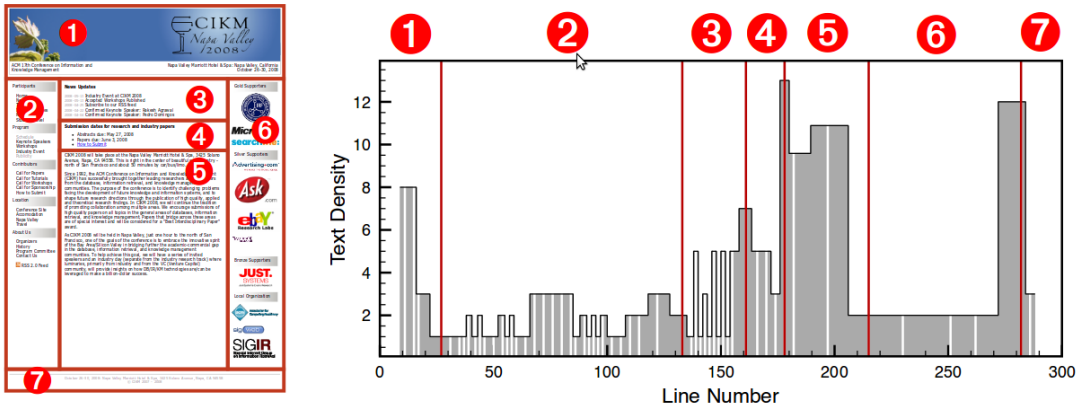

FIGURE 2 – Segmentation densitométrique [5]

1.3 Segmentation par indice visuel

2 Méta-modèle de page web

2.1 HTML 4

HTML 4 [1] 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'attributs prédéfinis qui décrivent l'objet inclus dans la page. Le principal étant *type* décrivant le type de contenu des données (*e.g.* figure 3). La valeur de ces attributs n'est pas prédéfinie. Elle est interprétée librement par la machine qui charge la page web.

```
<object data="data/test.mpg" type="video/mpeg">
        Ceci est une vidéo
</object>
```

FIGURE 3 – 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> et de leurs attributs respectifs : *id* et *class*.

DIV Signifiant division, la balise DIV est utilisée 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 quelconques décidées arbitrairement par l'auteur du document.

Les éléments DIV utilisés conjointement avec les attributs *id* et *class* sont au cœur du mécanisme générique de structuration d'un document. DIV permet de diviser le contenu d'un document en sections et sous-sections (*e.g.* figure 5) pour décrire sa structure. Les balises <DIV> ayant une sémantique neutre, c'est l'auteur du contenu qui attribue (de manière arbitraire) un nom de *class* ou un *id*. L'*id* ou la *class* est associé à une mise en forme définie a priori. La mise en forme est définie au travers d'un langage : CSS[2] que l'on appelle feuille de style. CSS permet d'appliquer un ensemble de règles de style ou un agencement des éléments dans l'espace de la page. Par exemple, l'auteur peut déclarer une classe "aside" et définir que les éléments appartenant à la classe "aside" doivent être placés sur le côté droit de la page avec un fond blanc. Ce mécanisme est illustré par la figure 4. L'auteur associe à chaque <DIV> une *class* ou un *id* auquel s'applique une mise en page et une mise en forme définies par l'auteur dans une feuille de style CSS.

Méta-modèle La figure 6 modélise les éléments principaux de construction d'une page web suivant la spécification de HTML 4.
- Chaque élément hérite de la méta-classe **Element**. Cela permet d'associer un identifiant unique à chaque élément ainsi qu'un ensemble de classe.

```
<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 4 – Architecture page web HTML 4

- Les éléments de la méta-classe **Bloc** sont l'ensemble des éléments dans une page qui forment un bloc structurel (*e.g.* un paragraphe).
- Les éléments de la méta-classe **InLine** définissent des éléments qui ne forment pas de blocs structurels. Par exemple la balise *strong* ne définit pas un bloc structurel de la page mais indique que l'élément qu'elle encapsule est un mot important dans un texte. De manière générale les éléments *InLine* servent à attribuer une sémantique aux éléments textuels.
- La relation de composition entre les éléments *Bloc*, spécifie que les éléments *Bloc* peuvent contenir des éléments *InLine* mais pas l'inverse. La relation de composition réflexive de l'élément *Bloc* spécifie qu'un élément en bloc peut contenir d'autre élément *Bloc*, il en est de même pour les éléments *InLine*.
- Les éléments de la méta-classe **Header** spécifient des éléments de titre. Ils introduisent le titre d'une section ou sous-section.
- Les éléments de la méta-classe **Div** spécifient une division structurelle : une section, sous section. En pratique elle est également utilisée pour signaler des regroupements d'élément

```
<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 5 – Exemple découpage en sections et sous-sections

afin de leurs appliquer une mise en forme.
- Les éléments de la méta-classe **Paragraph** forment une composition logique d'éléments textuels
- Les éléments de la méta-classe **Object** spécifient l'inclusion d'un contenu multimédia ou un programme.

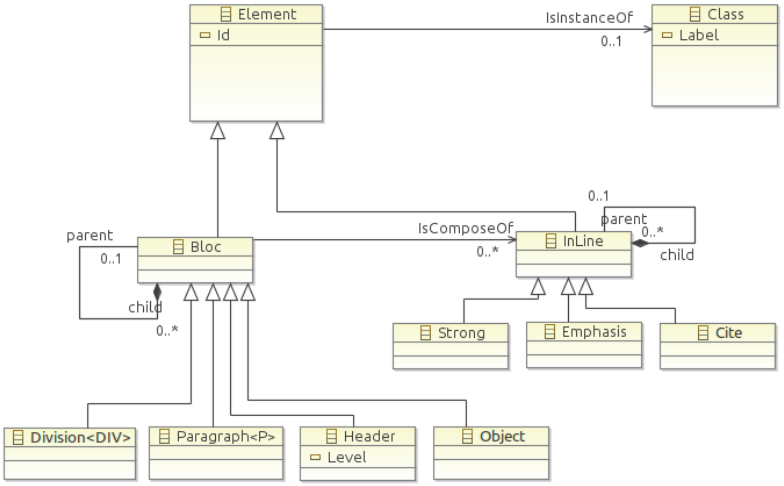

FIGURE 6 – Méta-modèle HTML 4

2.2 HTML 5

HTML 5 [4] étend HTML 4 en apportant de nouveaux éléments lexicaux. Ces nouveaux éléments apportent une sémantique standard et de plus haut niveau. Elle permet notamment d'expliciter la structure d'une page.

Contenu HTML 5 fournit de nouveaux éléments comme <VIDEO>, <AUDIO> avec un ensemble d'attributs propres à chaque balise (a 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 en opposition au découpage implicite de HTML 4 montré dans la figure 4.

Méta-modèle La figure 8 modélise les concepts principaux de construction d'une page web avec HTML 5.

- La méta-classe **Sectioning** définit le contenu comme des éléments qui créent une nouvelle section dans le plan d'un document. Ils définissent également la portée des éléments d' en-tête (Header) et de pied de page (Footer). Elle encapsule les concepts de division de HTML 4 (<DIV>).
- La méta-classe **Header** définit le contenu comme des éléments d'introduction. Par exemple pour un page web, un logo ou pour une section, un titre. Les sous classe de Header sont des éléments de titre introduisant des sections, ce sont les mêmes concepts de titre que pour HTML 4.
- La méta-classe **Footer** définit le contenu comme étant des éléments de pied de page ou de section.
- La méta-classe **Phrasing** définit le contenu textuelle, elle encapsule le concept de balise *Inline* de HTML 4.
- La méta-classe **Embedded** définit un contenu importé dans une page. C'est le cas par exemple des éléments de type <video> qui sont sous classe de la méta-classe Embedded (elle est également sous-classe de la méta-classe Interactive). Elle encapsule les concepts de *Object* de HTML 4.
- La méta-classe **Interactive** définit le contenu interactif dans une page. Par exemple, la balise <a> qui définit une navigation vers une autre ressource.

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

FIGURE 7 – Exemple d'attribution de rôle

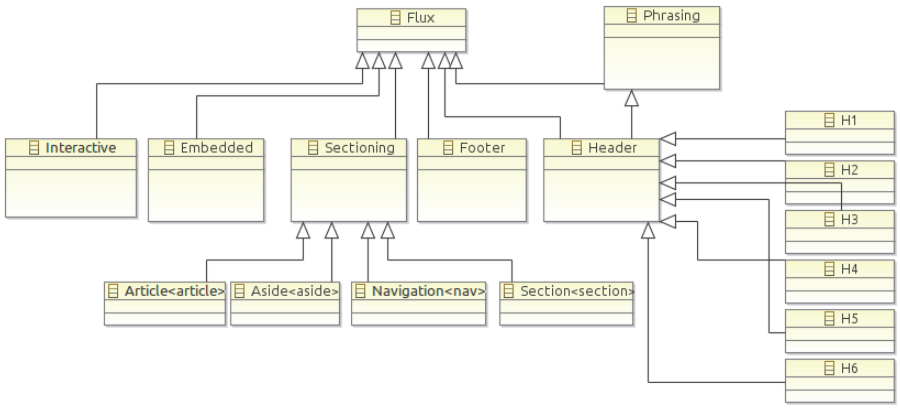

FIGURE 8 – Méta-modèle HTML 5

2.3 ARIA

ARIA (Acessible Rich Internet Application) [3] est la spécification d'une Ontologie décrivant une interface graphique. Elle fournit des informations sur la structuration d'un document et plus généralement elle 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 : définit un ensemble de widget (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.)

États 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 cochée, *false* si elle ne l'est pas et *mixed* dans le cas d'un ensemble de checkbox indique que certaines sont cochées.

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

Méta-modèle La figure **??** modélise les principales méta-classe de la norme ARIA. On distingue deux groupes. Un premier groupe pour la description des éléments d'interactions qui hérite de la méta-classe **Widget** :

- La méta-classe **Composite** indique qu'un élément graphique possède des éléments navigables. Elle permet d'organiser la navigation à l' intérieure d'un *Wigdet*.
- La méta-classe **Input** définit des éléments qui permettent des saisies de la part d'un utilisateur.
- La méta-classe **Command** définit des éléments qui réalisent des actions. Par exemple, l'envoie de données vers un serveur.

Un deuxième groupe pour les éléments de structuration qui héritent de la méta-classe **Structure**

- La méta-classe **Section** définit les éléments qui définissent une unité de confinement structurelle dans une page.
- La méta-classe **Sectionhead** définit les éléments qui introduisent des titres.
- La méta-classe **Landmark** définit les principaux points d ?intérêts dans une page. Par exemple, la banniere d'une page, les formulaires, le contenu principal.

2.4 CSS

CSS est un langage de feuille de style qui permet aux auteurs des pages web de lier du style aux éléments HTML. Le style définit comment afficher un élément (ex. les polices de caractères,

l'espacement, couleurs, *etc.*). CSS permet ainsi de séparer la présentation du style du contenu (*cf.* figure 9). L'avantage est une simplification de l'édition et de la maintenance d'une page web.

```
<style>
p.serif{font-family:"Times New Roman",Times,serif;}
p.sansserif{font-family:Arial,Helvetica,sans-serif;}
</style>

<body>
<p class="serif">Ceci est un paragraphe
avec un style de font Times New Roman font.</p>
<p class="sansserif">Ceci est un paragraphe
avec un style de font the Arial font.</p>
</body>
</html>
```

Ceci est un paragraphe avec un style de font Times New Roman font.

Ceci est un paragraphe avec un style de font the Arial font.

FIGURE 9 – Exemple CSS

Modèle de boîte CSS génère pour chaque élément de l'arbre du document (DOM) une boîte rectangulaire. Les boîtes rectangulaire sont conformes à un modèle de boîte et sont agencées suivant un modèle de mise en forme décrit en section 2.4

Chaque boîte possède ainsi une aire de contenu (*e.g* une texte, une image, *etc.*) entourée en option par une aire d'espacement, une aire de bordure et une aire de marge (*e.g* figure 10).

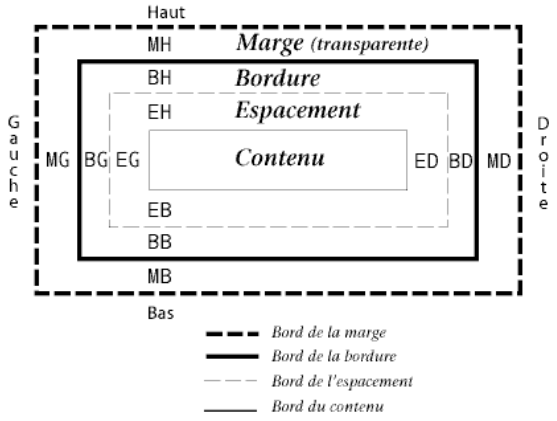

FIGURE 10 – Modèle de boîte

Modèle de mise en forme Chaque boîte se voit attribuer un type qui affecte en partie son positionnement. Les deux principaux types sont les *boîtes en bloc* et *les boîte en-ligne*. Les éléments

de type bloc sont des éléments dont le rendu visuel forme un bloc (*e.g* figure 11 avec l'élément de paragraphe <p>). Les éléments de type en-ligne sont des éléments qui n'ont pas la forme de blocs de contenu (*e.g* figure 11 avec l'élément). Les boîtes en-ligne sont placées horizontalement, l'une après l'autre, en commençant en haut du bloc conteneur. Les blocs conteneurs sont des boîtes qui encapsulent d'autres boîtes. Les boîtes en-bloc sont placées l'un après l'autre, verticalement, en commençant en haut du bloc conteneur. Le schéma de positionnement décrit est appelé *flux normal*.

```
<p>Avant de faire le truc X il est
<strong>nécessaire</strong> de faire le truc Y avant.
</p>
```

FIGURE 11 – Exemple élément en-line

Une fois le *flux normal* calculé, il est possible de le modifier.

Un premier mécanisme possible est le positionnement relatif. La position de la boîte est exprimée en propriété de décalage par rapport à son bloc conteneur :

- 'top' : définit le décalage du bord haut de la marge d'une boîte sous le bord haut de la boîte du bloc conteneur.
- 'right' : définit le décalage du bord droit de la marge d'une boîte à gauche du bord droit de la boîte du bloc conteneur.
- 'bottom' : définit le décalage du bord bas de la marge d'une boîte au-dessus du bord bas de la boîte du bloc conteneur.
- 'left' : définit le décalage du bord gauche de la marge d'une boîte à droite du bord gauche de la boîte du bloc conteneur.

Un deuxieme mécanisme est le positionnement flottant. Une boîte flottante est déplacée vers la gauche ou la droite sur la ligne courante du *flux normal*. Le contenu du document s'écoule alors le long des flancs de cette dernière.

Un troisième mécanisme est le positionnement absolu. La boîte est retirée du *flux normal* et est positionnée par rapport à son bloc conteneur. La différence avec le positionnement relatif est que le positionnement de la boîte n'a aucun effet sur les boîtes du même niveau de parenté. Ces boîtes peuvent, ou non, cacher les autres boîte.

Avant-plan et d'arrière-plan Les propriétés CSS permettent aux auteurs la spécification d'une couleur d'avant-plan et d'arrière-plan pour un élément. La couleur d'arrière-plan peut être une couleur ou une image . L'arrière-plan correspond aux aires de contenu et, d'espacement et de bordure. Le couleur d'avant-plan correspond à la couleur du contenu de texte d'un élément.

Les polices CSS permet de pouvoir spécifier l'utilisation de plusieurs représentation pour les caractères textuelles : *la police*. Une liste exaustive de propriétés permettent de spécifier la police d'un élément contenu dans une boîte. On peut spécifier par exemple une famille de police (serif, sans-serif), le style de la police (italic, oblique), la taille, *ect.*

Les textes CSS définies la représentation visuelle des caractères, des caractères blancs, des mots et des paragraphes. On peut spécifié un alinéa pour la première ligne du texte dans un bloc ('text-

indent'), l'alignement d'un contenu en-ligne dans un élément de type bloc ('text-align'), le comportement de l'espacement entre les caractères du texte ('letter-spacing'), *ect.*

3 Réalisation

3.1 Méta-modèle

3.1.1 Introduction

Le langage de publication de contenu actuel (HTML 4) fournit une composante syntaxique extrêmement souple. Il permet la construction d'un ensemble potentiellement infini de structures logiques desquelles on peut associer une même sémantique. Par exemple, la figure 12 présente deux structures logiques extraites d'une des pages du site web *lemonde.fr* et d'une page des pages du site web *www.eclipse.org*. D'autre part, HTML fournit une sémantique pour le contenu mais pas pour les structures logiques. D'après la figure 12, on sait que les éléments textuelles sont des liens de navigation, signalés par la balise *<a>*, mais il n'y a pas d'information sémantique sur la structure qu'ils composent (une menu de navigation).

La contribution de ce méta-modèle vient apporter une composante sémantique aux langages de publication du web. Cette sémantique s'intéresse à la correspondance entre les entités définies par le concepteur d'une page web et les entités comprissent par le lecteur. Elle est essentielle pour l'expression des souhaits de personnalisation de l'utilisateur. En second lieu, le méta-modèle nous fournit une couche d'abstraction permettant de s'affranchir de la diversité de représentation des données. Puis un séparation strict en les éléments de mise en forme et de structuration.

Le méta-modèle se décompose en deux parties. Une partie décrivant la sémantique des principales structures logiques utilisées dans la conception des pages. Une seconde partie pour la descriptions de la mise en forme associée à chaque élément du méta-modèle précédent. Cela va nous permettre de mieux intégrer l'accessibilité dans la personnalisation des pages web.

```
<div id="header-nav">
<ul>
        <li><a>Home</a></li>
        <li><a>Downloads</a></li>
        <li><a>Users</a></li>
        <li><a>Members</a></li>
        <li><a>Committers</a></li>
        <li><a>Resources</a></li>
        <li><a>Projects</a></li>
        <li><a>About Us</a></li>
</ul>
</div>
```

```
<div>
<p>
        <a>Le Monde</a><a>Télérama</a>
        <a>Le Monde diplomatique</a>
        <a>Le Huffington Post</a>
        <a>Courrier international</a>
        <a>La Vie</a>
        <a>au Jardin</a>
</p>
</div>
```

FIGURE 12 – Exemple de différentes conceptions de menu avec HTML4

3.1.2 Modèle de contenu

Chaque élément va nous permettre de qualifier les éléments d'une page. Une première partie décrie les éléments de structuration. Les principales méta-classes sont :

- Les méta-classes *SECTION*, elles définissent un environnement aux éléments relatifs d'un documents (titres, pieds de page, en-têtes). Un titre déclaré dans un section se rapporte à la déclaration d'une élément de type *SECTION* la plus proche. Ce concept d'environnement est introduit dans HTML5 par la balise de type <SECTION>.
- Les méta-classes *SECTIONHEAD* sont les éléments qui synthétisent un contenu. Il possède une portée locale, ceux-ci sont associées à l'élément de type *SECTION* le plus proche. Cette méta-classe englobe par exemple les balises de titres définit dans HTML4 (*e.g* <H1 H6>).
- Les méta-classes *REGION* correspondent à un regroupement par thématique des éléments qu'elles encapsulent, c'est à dire des éléments que l'on peut regrouper sur la base d'une information commune. Ce concept de regroupement par thématique est introduit dans HTML5 par la balise de type <SECTION>.
- Les méta-classes *LANDMARK* recensent les principales structures que l'on trouve dans une page web. Par exemple, la méta-classe *NAVIGATION* pour les menus (correspond à la balise de type *NAV* dans HTML 5) ou la méta-classe *BANNER* pour la bannière d'une page web.

Un seconde partie de ce méta-modèle décrit les éléments d'interaction dans une page web. Les principales méta-classes sont :

- Les méta-classe *INPUT* sont les éléments qui permettent des entrées utilisateurs. Elles correspondent aux balises de type <*input*> de HTML4.
- Les méta-classe *COMMAND* sont les éléments qui exécutent des actions mais ne reçoivent

pas d'information en entrée. Par exemple les balises de type <*a*> déclenche une action de navigation ou une balise de type <*button*> déclenche l'envoie de données d'un formulaire vers un serveur.
– Les méta-classe *SELECT* sont les éléments qui permettent de faire des sélections parmi un ensemble de choix.

3.1.3 Modèle de mise en forme

Le méta-modèle de mise en forme (*cf.* figure 15) est un support essentiel à l'expression des préférences, en particulier pour les préférences liées à l'accessibilité. Les principales méta-classe sont :

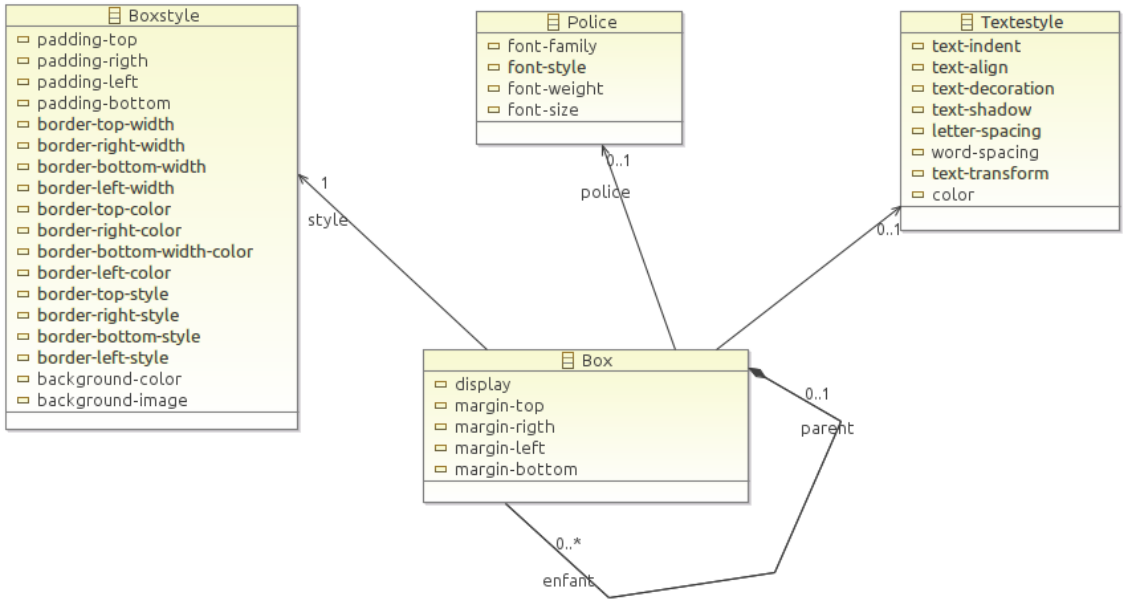

FIGURE 15 – Méta-modèle

– Les méta-classes *Box* qui représentent le concept de bloc conteneur. Elles vont permettre d'exprimer des préférences sur l'accessibilité tels que l'espacement plus ou moins fort (propriété padding) avec les éléments connexes à cette dernière. Mais encore sur les contrastes avec la propriété de couleur d'arrière plan et la propriété de couleur de la méta-classe *text*.
– Les méta-classes *Text* décrivent la représentation visuelle des caractères, mots et paragraphes contenus dans une boîte. Par exemple, la propriété *letter-spacing* spécifie l'espace entre les lettres.
– Les méta-classes *Méta-classe Police* décrivent la représentation visuelle des caractères. Dans le contexte de l'amélioration de l'accessibilité il être intéressant, par exemple, d'éviter certain type de font qui peuvent rendre un contenu plus illisible pour certaine pathologie.

3.2 Extraction des éléments du méta-modèle dans une page web

Les approches par d'extraction des éléments par comparaison de sous-structure dans des collections de documents ne permettent pas l'extraction de toute les structures dans une page, l'extraction est partielle.

Choix d'une segmentation descendante qui va permettre d'intégrer les nouvelles évolutions des langages de publication (HTML5 et RDFa).

Appendices

A Méta-modèle de Contenu

Widget Élément graphique dans une page web (bouton, liste déroulant, tableau, *etc.*). Il peut définir des éléments et un contenu interactif (*e.g* formulaire d'inscription, bar de recherche, fils d'actualité).

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Role
Sous Classe	Composite, Input, Command
Alignement HTML	

Composite Un élément composite est une composition d'éléments. Il représente un lien d' agrégation fort entre deux éléments, c'est à dire que si l'on supprime une classe agrégée la classe composite est détruite.

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Widget
Sous Classe	Select
Alignement HTML	

Input L'ensemble des éléments permettant des entrées utilisateurs.

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Widget
Sous Classe	Checkbox, Option, Select, Textbox
Alignement HTML	

Option Elément sélectionnable dans une liste.

Caractéristique	Valeur

Abstrait	Oui
Super Classe	Input
Sous Classe	Radio
Alignement HTML	<option>

Select Élément permettant à l'utilisateur de faire des sélections parmi un ensemble de choix.

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Composite, Group, Input
Sous Classe	Combobox, Listbox, RadioGroup
Alignement HTML	

Command Élément qui exécute des actions mais ne reçoit pas d'informations en entrée.

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Widget
Sous Classe	Bouton, Lien
Alignement HTML	

Button Élément graphique qui déclenche une action. Typiquement un bouton de validation d'un formulaire.

Caractéristique	Valeur
Abstrait	Non
Super Classe	Command
Sous Classe	
Alignement HTML	<button>

Link Définit une référence interactive vers une ressource interne ou externe à la page. Les navigateurs implémentent un comportement de navigation. Par exemple, une navigation vers une ressource interne peut être dans une page se déplacer de l'élément d'en-tête à l'élément de pied de page. Une navigation externe la page peut être un changement de page.

Caractéristique	Valeur
Abstrait	Non
Super Classe	Command
Sous Classe	
Alignement HTML	<a>, <link>

Checkbox Indique qu'une instance est cochable. Trois valeurs sont possibles (vraie-faux-mixte), indiquant si l'élément est coché ou non. La valeur mixte est utilisée dans le contexte d'un groupe

d'instance de type *checkbox*. Par exemple, lorsqu' il y' a au moins un élément coché et un non coché (*e.g* figure 16).

Caractéristique	Valeur
Abstrait	Non
Super Classe	Input
Sous Classe	Radiobox
Propriété	checked [vraie, faux, mixte]
Alignement HTML	<input type= 'checkbox' />

Radio Radio est une instance cochable. Il fait toujours partie d'une liste d'élément d'au moins deux éléments. Il présente la contrainte que l'on ne peut sélectionner qu'un seul élément parmi la liste de choix auquel il appartient (*e.g* figure 17).

Caractéristique	Valeur
Abstrait	Non
Super Classe	Checkbox, Option
Sous Classe	
Alignement HTML	☐

RadioGroup C'est une collection logique d'élément Radio. Dans html, la collection logique est exprimée par l'attribut name (*eg* figure 18)

Caractéritique	Valeur
Abstrait	Non
Super Classe	Select
Sous Classe	
Alignement HTML	

Listbox Wigdet qui permet de sélectionner un ou plusieurs élément dans une liste de choix(*e.g* figure 19).

Caractéristique	Valeur
Abstrait	Non
Super Classe	List, Select
Sous Classe	
Alignement HTML	<SELECT>

Combobox Élément qui permet de remplir un champ texte selon des options présentées dans une liste déroulante.

Caractéritique	Valeur
Abstrait	Non
Super Classe	Select
Sous Classe	
Alignement HTML	*cf.* figure 20

Structure Éléments de structuration dans une page web. Ce sont l'ensemble des éléments qui permettent d'organiser le contenu dans une page de manière logique.

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Roletype
Sous Classe	Section, Sectionhead, Separator, Texte
Alignement HTML	

List Les listes contiennent des éléments dont le rôle est *listitem* ou des éléments dont le rôle est *group* qui contiennent eux même des éléments *listitem* (*e.g* figure 22).

Caractéritique	Valeur
Abstrait	Non
Super Classe	région
Sous Classe	Listbox, Ol, Ul
Alignement HTML	

Ul (unorder list) Liste non ordonnée d'éléments (*e.g* figure 21).

Caractéritique	Valeur
Abstrait	Non
Super Classe	List
Sous Classe	
Alignement HTML	

OL(order list) Liste ordonnée d'éléments

Caractéritique	Valeur
Abstrait	Non
Super Classe	List
Sous Classe	
Alignement HTML	

Texte Elements qui définissent un état logique (sémantique) d'un texte, en opposition à un état physique (mise en forme). Cette partie se rajoute au méta-modèle de base de ARIA. Elle recense donc les éléments de HTML qui apportent un état logique. On exclut donc les éléments de mise en forme telles que <b> (bolt) qui traduisent un état physique (mise en forme) mais seront exprimés dans le méta-modèle de CSS.

Caractéristique	Valeur
Abstrait	Non
Super Classe	Structure
Sous Classe	Emphase, Abbreviation, Strong, Cite
Alignement HTML	

Emphase Mise en relief d'une partie du texte. Elle est généralement utilisé pour mettre en évidence un résumé d'article.

Abbreviation Définit une abreviation.

```
<p>Tony Blair est le premier
ministre de
la <abbr title="Grande-Bretagne">GB</abbr></p>
```

Strong Mot important dans un texte.

```
<p>Avant de faire le truc X
il est <strong>nécessaire</strong> de faire le truc
Y avant.</p>
```

Cite Elément de citation.

```
Ce référerer à la
norme <cite>[ISO-0000]</cite>
```

Section Aria définit une section comme une unité de confinement structurelle dans une page. On spécifie dans notre méta-modèle que les éléments héritant de section définissent des limites au contenu qu'il englobe. Ils fournissent un environnement contextuelle, c'est à dire une portée sémantique aux éléments. Par exemple, les éléments de titre se rapportent à l'élément de section qui l'a introduit.

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Structure
Sous Classe	Group, Region, Paragraph, Gridcell, Listitem
Alignement HTML	

ListItem Un élément dans une liste. Il est contenu dans une *listitem*.

Caractéritique	Valeur
Abstrait	Non
Super Classe	Section
Sous Classe	
Alignement HTML	

Group Élément regroupant une collection logique de widget (eg. figure **??**).

Caractéristique	Valeur
Abstrait	Non
Super Classe	Section
Sous Classe	Select
Alignement HTML	<fieldset>

Region Un groupement thématique dans une page. Éléments d'information sur une même thématique : représente une section générique d'un document. Fournit un environnement contextuel pour des éléments de titre, une entête et un pied de page.

Caractéritique	Valeur
Abstrait	Non
Super Classe	Section
Sous Classe	Article, Landmark, List
Alignement HTML	<section>

Paragraphe Élément rajouter au méta-modèle de aria. Définit une composition d'élément textuelle comme étant un paragraphe dans une page.

Caractéritique	Valeur
Abstrait	Non
Super Classe	Section

Caractéritique	Valeur
Sous Classe	
Alignement HTML	<p>

Gridcell Cellule d'une élément *Gridcell* (*eg.* figure **??**).

Caractéristique	Valeur
Abstrait	Non
Super Classe	Section
Sous Classe	
Alignement HTML	<td>

Rowheader Une cellule contenant des informations d'entête pour une ligne de *Gridcell* (*eg.* figure **??**).

Caractéritique	Valeur
Abstrait	Non
Super Classe	Section
Sous Classe	
Alignement HTML	<th>

Row Une ligne dans un *Gridcell* (*eg.* figure **??**).

Caractéritique	Valeur
Abstrait	Non
Super Classe	Group
Sous Classe	
Alignement HTML	<tr>

Grid Aria spécifie cet élément comme un contenu interactif car il permet d'organiser la navigation dans les données qu'il structure. Il peut spécifier notamment un agencement structuré de l'interface graphique d'une page. Nous réduisons la norme Aria de *Grid* comme un élément qui contient des données tabulaires organisées en ligne et colonne (*eg.* figure **??**).

Caractéritique	Valeur
Abstrait	Non
Super Classe	Region
Sous Classe	
Alignement HTML	<table>

Separator Élément qui marque une division dans le contenu d'une section. Il permet de mieux signaler les contenus sémantiquement différents. Ce sont des séparateurs visuelles (lignes de pixels vides horizontales ou verticales entre deux éléments).

Caractéritique	Valeur
Abstrait	Non
Super Classe	Structure
Sous Classe	
Alignement HTML	<hr>

SectionHead Élément qui résume ou décrive brièvement le sujet introduit par une région.

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Structure
Sous Classe	Heading
Alignement HTML	

Heading Définit un élément titre

Caractéristique	Valeur
Abstrait	Oui
Super Classe	Sectionhead
Sous Classe	
Alignement HTML	<H1,2,3,4,5,6>

Landmark Éléments structurels courant dans une page web.

Caractéritique	Valeur
Abstrait	Oui
Super Classe	Region
Sous Classe	Banner, Main, Form, Navigation, Complementary, ContentInfo, Application
Alignement HTML	

Banner Pour faire l'analogie avec l'entête d'un document, on parle de bannière pour une page web.

Caractéritique	Valeur
Abstrait	Non
Super Classe	Landmark
Sous Classe	
Alignement HTML	<header>

Application Contenu applicatif dans la page.

Caractéritique	Valeur
Abstrait	Non
Super Classe	Landmark
Sous Classe	Audio, Video
Alignement HTML	<header>

Complementary Région d'un document conçut comme étant complémentaire au contenu principal du document auquel il est associé. Il conserve une signification même si il est séparer du contenu principale.

Caractéritique	Valeur
Abstrait	Non
Super Classe	Landmark
Sous Classe	
Alignement HTML	<ASIDE>

ContentInfo Région d'un document contenant des informations sur celui-ci. Par exemple le copyrights associé à un document.

Caractéritique	Valeur
Abstrait	Non
Super Classe	Landmark
Sous Classe	
Alignement HTML	

Form Région qui contenant une collection d'éléments formant un formulaire. Les éléments sont généralement une collection de *command, input* qui permettent une interaction avec l'utilisateur. Les interactions permettre d'envoyer des informations à un agent en vu d'un traitement (*cf.* figure **??**).

Caractéristique	Valeur
Abstrait	Non
Super Classe	Landmark
Sous Classe	
Alignement HTML	<form>

Main Le contenu principale dans une page.

Caractéristique	Valeur
Abstrait	Non
Super Classe	Landmark
Sous Classe	

Alignement HTML	

Navigation Région contenant une collection de lien navigable vers des ressources internes ou externes. Par exemple le menu de navigation d'une page web (*cf.* figure **??**).

Caractéritique	Valeur
Abstrait	Non
Super Classe	Landmark
Sous Classe	
Alignement HTML	<nav>

B Méta-modèle de mise en forme

Méta-classe Box La méta-classe *Box* décrit les propriétés de positionnement des éléments. Les différentes propriétés sont :

- display : sert à définir le schéma de positionnement appliqué à la boîte. Les deux principaux étant *inline* et *block*. inline positionne les éléments sur la même ligne alors que *block* positionne les éléments les un sous les autres.
- margin (top, left, right, bottom) : spécifie l'espacement du bord extérieur de la boîte.

Méta-classe Style La méta-classe *Style* décrit les boîtes rectangulaires qui sont générées pour les éléments de l'arbre du document (*cf.*figure **??**). Les différentes propriétés sont :

- padding : l'air d'espacement (*padding*)
- border-width épaisseur de bordure
- *border-style* : style de la bordure
- border-color : la couleur de bordure
- border-[color, image] : arière-plan

Méta-classe Text La méta-classe *Text* décrit la représentation visuelle des caractères, mots et paragraphes contenu dans une boîte. Les différentes propriétés sont :

- text-indent : décrit un alinéa
- text-align : décrit un alignement. Exemple de valeur possible : alignement de texte à gauche, droite, centré, *ect.*
- decoration : décrit un trait en-dessous, trait au-dessus, rayure et clignotement
- text-shadow : décrit des effets d'ombrage appliquer au texte
- letter-spacing : décrit l'espacement entre les mots
- word-spacing : décrit l'espacement entre les mots
- text-transform : décrit les effets de capitalisation dans le texte. Par exemple la valeur *uppercase* définit que les lettres de chaque mots soient en majuscule, *lowercase* décrit l'inverse.
- color : décrit la couleur du texte

Méta-classe Police La méta-classe *Police* décrit la représentation visuelle des caractères :

- font-family : décrit les noms de famille générique de la police du texte (*e.g new century schoolbook*)

- font-style : style de la police (*e.g italic*)
- font-weight : décrit la graisse de la police
- font-size : décrit la taille de la police

Glossaire

Ontologie En philosophie, l'ontologie est l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe. Par analogie, le terme est repris en informatique et en science de l'information, où une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations [?]. 12

Références

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

[2] World Wide Web Consortium et al. Cascading Style Sheets. http ://www.w3.org/Style/CSS/, 2010.

[3] World Wide Web Consortium et al. Accessible Rich Internet Applications 1.0. http ://www.w3.org/WAI/intro/aria, 2014.

[4] World Wide Web Consortium et al. HTML 5 Specification. http ://www.w3.org/TR/html5/, 2014.

[5] Christian Kohlschütter and Wolfgang Nejdl. A densitometric approach to web page segmentation. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1173–1182. ACM, 2008.

[6] Milos Kovacevic, Michelangelo Diligenti, Marco Gori, and Veljko Milutinovic. Recognition of common areas in a web page using visual information : a possible application in a page classification. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 250–257. IEEE, 2002.

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

[8] Karane Vieira, Altigran S da Silva, Nick Pinto, Edleno S de Moura, Joao Cavalcanti, and Juliana Freire. A fast and robust method for web page template detection and removal. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 258–267. ACM, 2006.

[9] Jason TL Wang and Kaizhong Zhang. Finding similar consensus between trees : an algorithm and a distance hierarchy. *Pattern Recognition*, 34(1) :127–137, 2001.

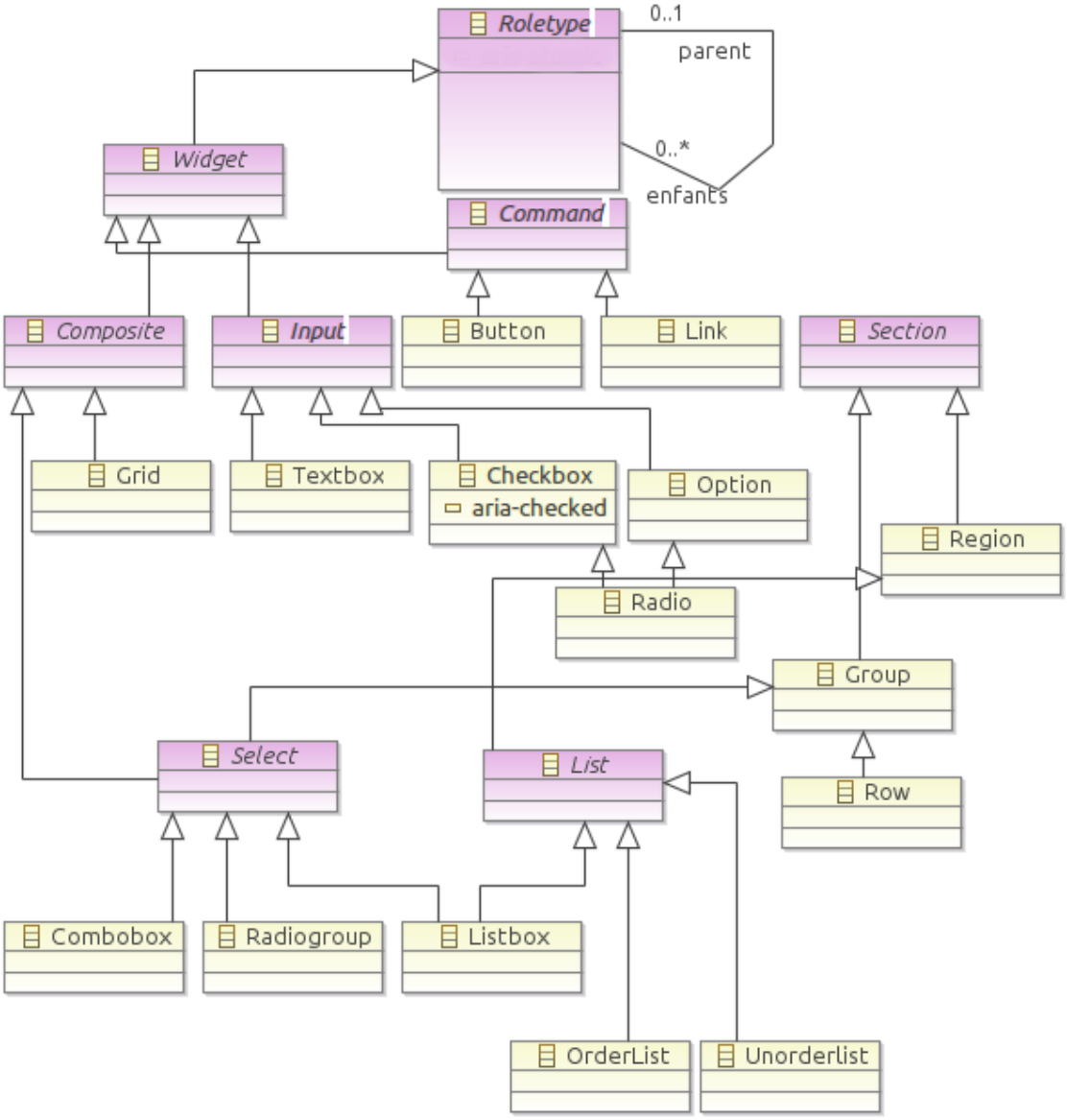

FIGURE 13 – Méta-modèle

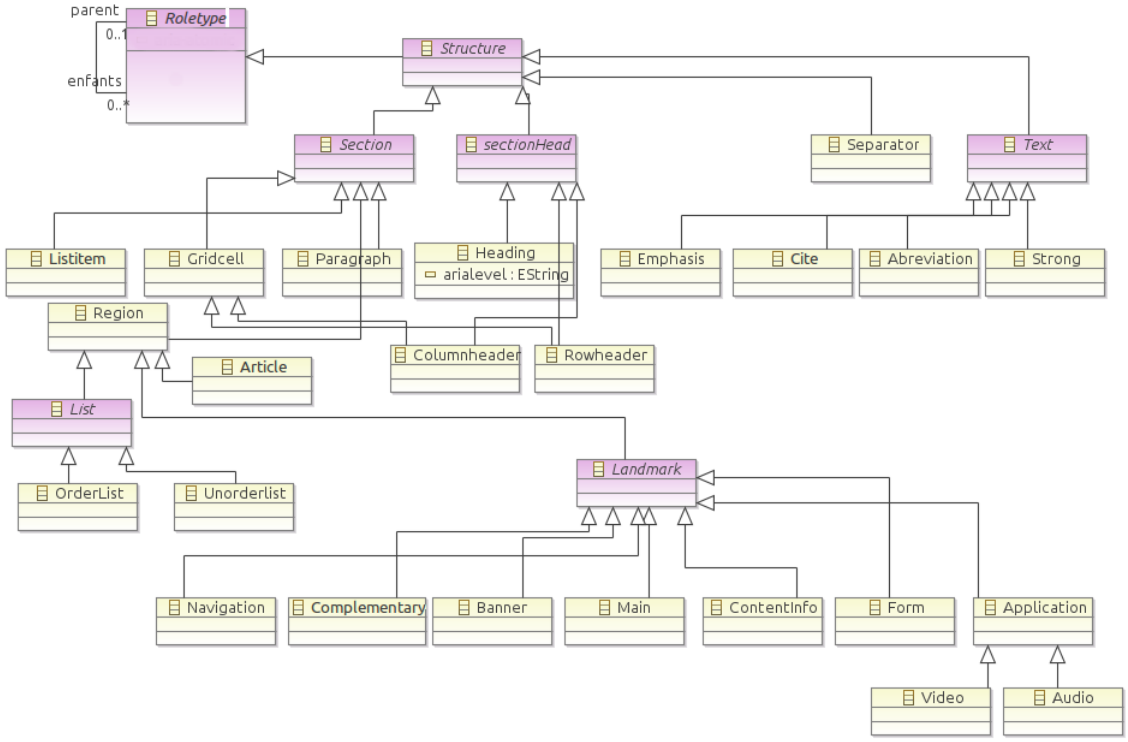

FIGURE 14 – Méta-modèle

```
<input type="checkbox">I have a bike<br>
<input type="checkbox">I have a car
```

FIGURE 16 – Exemple de checkbox

```
<input type="radio">Male<br>
<input type="radio">Female
```

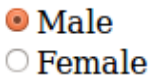

FIGURE 17 – Exemple de radio

```
<input type="radio" name="vin">rouge
<input type="radio" name="vin">blanc
<input type="radio" name="vin">rose
```

FIGURE 18 – Exemple de Radiogroup

```
<select>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```

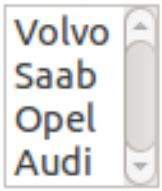

FIGURE 19 – Exemple de Select

```
<input type=text list=browsers >
<datalist id=browsers >
   <option> Google
   <option> IE9
   <option> Firefox
</datalist>
```

FIGURE 20 – Exemple de Combobox

```
<ul>
  <li>Cafe</li>
  <li>Thé</li>
  <li>Lait</li>
</ul>
```

FIGURE 21 – Exemple de liste non-ordonnée

```
<ol>
  <li>Cafe</li>
  <li>Thé</li>
  <li>Lait</li>
</ol>
```

1. Cafe
2. Thé
3. Lait

FIGURE 22 – Exemple de liste ordonnée